

## **Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones**



### **Trabajo Fin de Máster**

## **Identificación de rocas a partir de imágenes obtenida desde dispositivos móviles.**

Autor: Eduardo Méndez Rodríguez  
Tutor/es: Carlos Manuel Travieso González  
Fecha: 19 de Enero de 2018



# Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones

## Hoja de Evaluación del TFM



## Identificación de rocas a partir de imágenes obtenida desde dispositivos móviles.

Calificación: .....

Presidente

Secretario

Vocal

Fdo.:

Fdo.:

Fdo.:

Fecha: 19 de Enero de 2018



## ÍNDICE DE CONTENIDOS

<b>CAPÍTULO I: ESTADO DEL ARTE</b>	<b>15</b>
1.1. INTRODUCCIÓN	17
1.2. ESTADO DEL ARTE	18
1.3. OBJETIVOS DE LA PROPUESTA	20
1.4. ESQUEMA DE LA MEMORIA	21
<b>CAPÍTULO II: BASE DE DATOS</b>	<b>23</b>
2.1. INTRODUCCIÓN	25
2.2. CONSTRUCCIÓN DE LA BASE DE DATOS	25
2.2.1. Toma de datos en campo.	25
2.2.2. Localización de las tomas.	25
2.2.3. Características de la base de datos	45
<b>CAPÍTULO III: MÉTODOS APLICADOS</b>	<b>47</b>
3.1. INTRODUCCIÓN	49
3.2. MÉTODOS DE EXTRACCIÓN DE CARACTERÍSTICAS	49
3.2.1. LOCAL BINARY PATTERN (LBP)	49
3.2.2. DISCRETE WAVELET TRANSFORM (DWT)	51
3.2.2.1. DWT 2D	51
3.2.2.2. Arquitectura de aproximación directa.	53
3.3. REDES NEURONALES ARTIFICIALES O ARTIFICIAL NEURONAL NETWORK (ANN)	56
3.3.1. Red neuronal Perceptrón	58
3.3.2. Red neuronal Perceptrón Multicapa	58
<b>CAPÍTULO IV: METODOLOGIA EXPERIMENTAL Y RESULTADOS</b>	<b>60</b>
4.1. INTRODUCCIÓN	62
4.2. METODOLOGÍA	62
4.2.1. Base de datos a escala de grises.	63
4.2.2. Primer experimento: Clasificación por LBP.	64
4.2.3. Segundo experimento: Extracción de LBP por nivel de DWT	69
4.2.4. Tercer experimento: DWT + LBP conjuntamente	72
4.3. RESULTADOS	76
4.3.1. Experimento 1: LBP	76
4.3.2. Experimento 2: Extracción de LBP por nivel de DWT	81
4.3.3. Experimento 3: DWT + LBP conjuntamente	83
4.4. COMPARATIVA CON EL ESTADO DEL ARTE	89
<b>CAPÍTULO V: CONCLUSIONES Y LINEAS FUTURAS</b>	<b>91</b>

---

---

<b>5.1. CONCLUSIONES</b>	<b>93</b>
<b>5.2. LINEAS FUTURAS</b>	<b>94</b>
<b>5.3. REFERENCIAS BIBLIOGRÁFICAS</b>	<b>95</b>
<b>ANEXO A: RESUMEN DEL ESTADO DEL ARTE</b>	<b>99</b>
<b>ANEXO B: ALGORITMOS EMPLEADOS</b>	<b>107</b>
Algoritmo para pasar imágenes a escala de grises.	109
Algoritmo para obtener LBP	110
Algoritmo creado para el primer experimento	116
Algoritmo creado para el segundo experimento	120
Algoritmo creado para el tercer experimento	122

## ÍNDICE ILUSTRACIONES

<i>Ilustración 2. 1. Ubicación de las muestras del tipo A</i> .....	26
<i>Ilustración 2. 2. Muestra del tipo de roca A</i> .....	27
<i>Ilustración 2. 3. Ubicación de las muestras del tipo B</i> .....	27
<i>Ilustración 2. 4. Muestra del tipo de roca B</i> .....	28
<i>Ilustración 2. 5 Ubicación de las muestras del tipo C</i> .....	28
<i>Ilustración 2. 6. Tipo de rocas a fotografiar del tipo C</i> .....	28
<i>Ilustración 2. 7. Muestra del tipo de roca C</i> .....	29
<i>Ilustración 2. 8 . Ubicación de las muestras del tipo D</i> .....	29
<i>Ilustración 2. 9. Muestra del tipo de roca D</i> .....	30
<i>Ilustración 2. 10. Ubicación de las muestras E, F y G</i> .....	30
<i>Ilustración 2. 11. Tipo de rocas a fotografiar del tipo E</i> .....	31
<i>Ilustración 2. 12. Muestra de tipo de rocas del tipo E</i> .....	31
<i>Ilustración 2. 13 Tipo de rocas a fotografiar del tipo F</i> .....	32
<i>Ilustración 2. 14. Muestra de tipo de rocas del tipo E</i> .....	32
<i>Ilustración 2. 15. . Tipo de rocas a fotografiar del tipo G</i> .....	32
<i>Ilustración 2. 16. Muestra de tipo de rocas del tipo G</i> .....	33
<i>Ilustración 2. 17. . Tipo de rocas a fotografiar del tipo H, I, J</i> .....	33
<i>Ilustración 2. 18. Muestra de tipo de rocas del tipo H</i> .....	34
<i>Ilustración 2. 19. Muestra de tipo de rocas del tipo I</i> .....	34
<i>Ilustración 2. 20. Muestra de tipo de rocas del tipo J</i> .....	35
<i>Ilustración 2. 21. Ubicación de las muestras del tipo K</i> .....	35
<i>Ilustración 2. 22. Tipo de rocas a fotografiar del tipo K</i> .....	36
<i>Ilustración 2. 23. Muestra de tipo de rocas del tipo K</i> .....	36
<i>Ilustración 2. 24 .Ubicación de las muestras del tipo L</i> .....	36
<i>Ilustración 2. 25. Tipo de roca L</i> .....	37
<i>Ilustración 2. 26. Muestra de tipo de rocas del tipo L</i> .....	37
<i>Ilustración 2. 27. Ubicación de las muestras LL y M</i> .....	37
<i>Ilustración 2. 28. Tipo de rocas a fotografiar del tipo LL y M</i> .....	38
<i>Ilustración 2. 29. Muestra de tipo de rocas del tipo LL</i> .....	38
<i>Ilustración 2. 30. Muestra de tipo de rocas del tipo M</i> .....	39
<i>Ilustración 2. 31. Ubicación de las muestras del tipo N</i> .....	39
<i>Ilustración 2. 32. Tipo de rocas a fotografiar del tipo N</i> .....	39
<i>Ilustración 2. 33. Muestra de tipo de rocas del tipo N</i> .....	40
<i>Ilustración 2. 34. Ubicación de las muestras del tipo O</i> .....	40
<i>Ilustración 2. 35. . Tipo de rocas a fotografiar del tipo O</i> .....	41
<i>Ilustración 2. 36. Muestra de tipo de rocas del tipo O</i> .....	41
<i>Ilustración 2. 37 Ubicación de las muestras del tipo P</i> .....	42
<i>Ilustración 2. 38. Tipo de rocas a fotografiar del tipo P</i> .....	42
<i>Ilustración 2. 39. Muestra de tipo de rocas del tipo P</i> .....	43
<i>Ilustración 2. 40 Ubicación de las muestras del tipo Q</i> .....	43
<i>Ilustración 2. 41. Tipo de rocas del tipo Q</i> .....	44

<i>Ilustración 2. 42. Muestra de tipo de rocas del tipo Q.....</i>	<i>44</i>
<i>Ilustración 2. 43. Características técnicas de la cámara del sensor de captura de imágenes.....</i>	<i>45</i>
<i>Ilustración 3. 1. Cálculo del LBP.....</i>	<i>50</i>
<i>Ilustración 3. 2. Análisis de los bordes, LBP.....</i>	<i>50</i>
<i>Ilustración 3. 3. Histograma del LBP.....</i>	<i>51</i>
<i>Ilustración 3. 4. Diagrama DWT.....</i>	<i>52</i>
<i>Ilustración 3. 5. Diagrama DWT linealmente no separable.....</i>	<i>53</i>
<i>Ilustración 3. 6. Esquema de DWT no separable, por arquitectura por aproximación directa, cuarto nivel.....</i>	<i>54</i>
<i>Ilustración 3. 7. Esquema de DWT de una imagen por aproximación directa.....</i>	<i>55</i>
<i>Ilustración 3. 8. Esquema de una Neurona.....</i>	<i>56</i>
<i>Ilustración 3. 9. Esquema de una ANN.....</i>	<i>58</i>
<i>Ilustración 3. 10. Esquema de una ANN.....</i>	<i>59</i>
<i>Ilustración 3. 11. Esquema de los experimentos realizados.....</i>	<i>63</i>
<i>Ilustración 4. 2. Esquema conversión a escala de grises.....</i>	<i>64</i>
<i>Ilustración 4. 3. Histograma de frecuencia LBP.....</i>	<i>65</i>
<i>Ilustración 4. 4. Extracto de etiquetas de salida, roca tipo B, simulación con 24 neuronas una única iteración, análisis de tipo roca B.....</i>	<i>66</i>
<i>Ilustración 4. 5. Extracto de etiquetas de salida, simulación con 24 neuronas una única iteración, análisis de tipo roca A.....</i>	<i>67</i>
<i>Ilustración 4. 6. Esquema del primer experimento.....</i>	<i>69</i>
<i>Ilustración 4. 7. Esquema del método DWT aplicando la arquitectura de aproximación directa hasta el 4º nivel.....</i>	<i>70</i>
<i>Ilustración 4. 8. Extracto de la matriz de parámetros, param2 (256x1140).....</i>	<i>70</i>
<i>Ilustración 4. 9. Matriz de etiqueta, target2 (19x1140).....</i>	<i>71</i>
<i>Ilustración 4. 10. Esquema del segundo experimento.....</i>	<i>72</i>
<i>Ilustración 4. 11. Extracto de vector LBP, concatenando los resultados LBP de GG, GG2, GG3 y GG4.....</i>	<i>72</i>
<i>Ilustración 4. 12. Matriz de parámetros, param (665x1024), para el tercer experimento.....</i>	<i>73</i>
<i>Ilustración 4. 13. Matriz de etiqueta, target (19x665), para el tercer experimento.....</i>	<i>73</i>
<i>Ilustración 4. 14. Filtro de Haar.....</i>	<i>74</i>
<i>Ilustración 4. 15. Filtro biortogonal 4.4.....</i>	<i>74</i>
<i>Ilustración 4. 16. Filtro Symlet 4.....</i>	<i>75</i>
<i>Ilustración 4. 17. Esquema del tercer experimento.....</i>	<i>75</i>
<i>Ilustración 4. 18. Gráfica de clasificación por nº de neuronas para el primer experimento.....</i>	<i>76</i>
<i>Ilustración 4. 19. Muestra de tipo 19 (izquierda) y 17 (derecha).....</i>	<i>77</i>
<i>Ilustración 4. 20. Gráfica del porcentaje de éxito de una red con 10 neuronas ocultas.....</i>	<i>78</i>
<i>Ilustración 4. 21. Gráfica del porcentaje de éxito de una red con 24 neuronas ocultas.....</i>	<i>78</i>

---

---

<i>Ilustración 4. 22. Gráfica del porcentaje de éxito de una red con 28 neuronas ocultas</i>	<i>_ 79</i>
<i>Ilustración 4. 23. Gráfica del porcentaje de éxito de una red con 30 neuronas ocultas</i>	<i>_ 79</i>
<i>Ilustración 4. 24. Mejor resultado por n<sup>o</sup> de iteraciones, tercer experimento.</i>	<i>_____ 84</i>
<i>Ilustración 4. 25. Muestra del tipo de roca n<sup>o</sup> 7.</i>	<i>_____ 86</i>
<i>Ilustración 4. 26. Muestra del tipo de roca n<sup>o</sup> 13.</i>	<i>_____ 86</i>
<i>Ilustración 4. 27. Muestra del tipo de roca n<sup>o</sup> 13.</i>	<i>_____ 87</i>

## ÍNDICE TABLAS

<i>Tabla 2. 1 Características técnicas de la cámara del sensor de captura de imágenes.....</i>	<i>45</i>
<i>Tabla 4. 1. Matriz de etiqueta, ejemplo 3 muestra por 3 tipos de rocas.....</i>	<i>65</i>
<i>Tabla 4. 2. Matriz de confusión de tipos de rocas, ejemplo explicativo de una matriz de confusión.....</i>	<i>67</i>
<i>Tabla 4. 3. Fórmula para obtener porcentaje de éxito a partir de la matriz de confusión .....</i>	<i>68</i>
<i>Tabla 4. 4. Resultado de aplicar fórmula de porcentaje de éxito a partir de la matriz de confusión de ejemplo.....</i>	<i>68</i>
<i>Tabla 4. 5. Matriz de confusión para una red con 24 neuronas, y una única iteración.....</i>	<i>68</i>
<i>Tabla 4. 6. Matriz de confusión de la etiquetasalida para una red con 28 neuronas, y 12 iteraciones, en el segundo experimento.....</i>	<i>71</i>
<i>Tabla 4. 7. Matriz de confusión del promedio de etiquetas de salidas, primer experimento.....</i>	<i>77</i>
<i>Tabla 4. 8. Matriz de confusión de etiquetas de salida, 24 neuronas con 10 iteraciones primer experimento.....</i>	<i>80</i>
<i>Tabla 4. 9. Porcentaje de éxito según nº de iteraciones, para 24 neuronas.....</i>	<i>80</i>
<i>Tabla 4. 10. Matriz de confusión promedio de etiquetas de salida 24 neuronas, primer experimento.....</i>	<i>81</i>
<i>Tabla 4. 11. Comparativa de resultados de éxito entre los dos primeros experimentos.....</i>	<i>82</i>
<i>Tabla 4. 12. Evaluación de las familias Wavelet.....</i>	<i>83</i>
<i>Tabla 4. 13. Resultados de porcentaje de éxito en el experimento tercero.....</i>	<i>84</i>
<i>Tabla 4. 14. Matriz de confusión de la etiqueta de salida 18 neuronas y 25 iteraciones, tercer experimento.....</i>	<i>85</i>
<i>Tabla 4. 15. Resumen de porcentaje de éxito de estado del arte.....</i>	<i>90</i>



**LISTADO DE ACRONIMOS:**

- ULPGC: Universidad de Las Palmas de Gran Canaria
- TFM: Trabajo Fin de Máster
- TIC: Tecnologías de la Información y la Comunicación
- BIMeTIC: Máster en soluciones TIC en el Bienestar y Medioambiente
- MATLAB: *MATrix LABoratory*
- ANN: *Artificial Neuronal Network*
- RNA: Redes Neuronales Artificiales
- SVM: *Support Vector Machine*
- MSV: Máquina de Soporte Vectorial
- LBP: *Local Binary Pattern*
- DWT: *Discrete Wavelet Transform*
- GG: Filtro DWT pasó Bajo-bajo
- GH: Filtro DWT pasó Bajo-alto
- HG: Filtro DWT pasó Alto-bajo
- HH: Filtro DWT pasó Alto-alto
- Haar: Transformada Haar Wavelet, primer Wavelet
- Bior4.4: Bio-ortogonal 4.4. de Wavelet
- Sym4: Symlet 4 de Wavelet
- I+D+i: Innovación, Desarrollo e investigación
- ID: Identification
- RICC: Representación de Imágenes por Cúmulos Coordinados
- CIELab: CIE L\*a\*b\*, espacio de color que puede percibir el ojo humano.
- Px: píxel
- Mpx: Megapíxel
- HD: *High definition*
- LED: *Light Emitting Diode*
- JPEG o JPG: *Joint Photographic Experts Group*
- cm: Centímetros
- ppp: puntos por pulgadas
- BIT: *Binary Digital* o Dígito Binario



# CAPÍTULO I: ESTADO DEL ARTE



## 1.1. INTRODUCCIÓN

A lo largo de la historia, la geología a tratado de comprender como ha evolucionado La Tierra, mediante ella, hoy en día es posible conocer los procesos geológicos que han sucedido en una determinada zona, como se han formado los accidentes geográficos, la edad de la tierra, entre otras muchas aportaciones.

El análisis de las rocas, es una de las primeras materias que aprende un geólogo, donde estudian sus características (físicas, químicas y mecánicas), existiendo principalmente tres métodos en la identificación de rocas, geoquímicos, macroscópicos y microscópicos [1].

Los métodos macroscópicos son directos, es decir, se usa la vista para identificarlos así como herramientas de apoyo sencillas (lupas, martillo, etc.). Este método lo emplean los geólogos para una valoración preliminar o definitiva en función de si dicha roca, es reconocible a simple vista, en caso contrario se utilizan el resto de métodos para esclarecer de qué roca se trata.

La evolución tecnológica y el avance de la inteligencia artificial, han propiciado que se combinen ambas ciencias para desarrollar herramientas que permitan la identificación automática de rocas. Estas herramientas han permitido trasladarlas al área de la construcción, industria, etc., una de las posibles utilidades es la selección por tamaño o tipo y separarlas de manera automática o servir de segunda opinión para experto en geología.

La evolución de los Smartphone ha permitido que se desarrollen aplicaciones como MediaMusea, esta permite fotografiar una obra de arte y darnos toda la información sobre ella.

Con todo lo anteriormente comentado, se propone por parte de un profesor de la universidad, realizar una aplicación similar pero en la identificación de rocas, previamente para poder desarrollar dicha aplicación es necesario una base de datos y un sistema de clasificación de imágenes de tipos de rocas, siendo este primer paso, el nacimiento del presente Trabajo Fin de Máster, en adelante TFM.

## 1.2. ESTADO DEL ARTE

A continuación se describe la evolución de los trabajos y proyectos realizados en la clasificación de rocas.

El primer trabajado encontrado en la identificación de rocas data de 1973 [2], realizado por *Robert M. Haralick, K. Shanmugam, y Its'hak Dinstein*, donde las muestras empleadas para la detección de rocas son de tipo areniscas, las cuales fueron obtenidas por fotomicrografía, imágenes aéreas e imágenes satelitales. Extrayendo las características de textura de las imágenes, para posteriormente aplicar la técnica de clasificación supervisada del vecino más próximo, donde se logró porcentajes de éxito del 89 %.

En 1985 se consigue el mismo porcentaje de éxito que en 1973, pero esta vez se analizan un total de seis tipos de rocas, donde se extraen las características texturales mediante la matriz de coocurrencia, el espectro y espaciado textural [23].

En 2001 [3] la universidad de Toronto en Canadá realiza un trabajo para la clasificación de minerales, donde las imágenes se obtienen en laboratorio mediante microscopio, estas se denominan como secciones finas, posteriormente se les aplica luz polarizada y se extraen las características de las imágenes originales y las polarizadas, tras esto se clasifican mediante redes neuronales, consiguiendo un porcentaje de éxito del 93 %.

En 2002 [5] se desarrolló un sistema de recuperación de imágenes por contenido mediante distintos programas (*MUVIS, QBIC and IMatch*), donde se clasificó un total de 7 tipos de rocas, extrayendo las características texturales, máscaras de filtrado y transformada de *Hough*, consiguiendo porcentajes de éxito del 65,1 %, 68,3 % y 63,5% respectivamente, posteriormente al combinar los datos anteriores el porcentaje ascendió al 81 %.

En 2005 se encuentran dos estudios realizados aplicando redes neuronales, donde se desean clasificar rocas marinas [6] y minerales enriquecidos [7] respectivamente, en ambos sistemas se empleó el software Matlab, obteniendo porcentajes de éxito del 93,5% para las imágenes marinas y del 88,71 % para los minerales enriquecidos.

En el III encuentro de la Mujer en la ciencia [8], desarrollado en 2006, se presenta un trabajo desarrollado por *J. Trinidad Guillen-Bonillaa y Evguenii Kurmyshev*, donde se desarrolló un método de evaluación de la adyacencia (semejanza) entre clases, a través de un clasificador en clases múltiples de distancia mínima en el espacio característico de la Representación de Imágenes por Cúmulos Coordinados (RICC). Clasificando un total de

8 muestras de imágenes de granito Rosa Porriño, donde se consigue un porcentaje de éxito del 100 % en la clasificación de las rocas ornamentales mediante la transformada de la RICC.

Las universidades de Turquía e India realizan en 2009 experimentos independientes en la identificación de rocas empleando redes neuronales. La universidad de Turquía realiza la clasificación de minerales extrayendo las características texturales de las secciones finas sobre luz polarizada cruzada, consiguiendo porcentajes de éxito del 81 al 98% [9]. Por otro lado se clasifica rocas basálticas extrayendo 27 características texturales con un éxito de clasificación del 92,22 % de éxito [10].

En 2011 la universidad de Chile [11] desarrolla un estudio sobre un trabajo anterior en la clasificación de una base de datos de rocas de cantera, para la clasificación en tiempo real, donde se extrajeron las características del color y textura, se subdividieron las imágenes y además se clasifican por separado las secciones finas de rocas húmedas y secas, aplicando la técnica de clasificación de máquinas de soporte vectorial, se consigue reducir el error mínimo residual en un 29,5%, con respecto a estudios anteriores sobre la misma base de datos.

Mediante el programa *TSecSoft* desarrollado en Matlab, se aplica sobre la base de datos del programa, pre-procesado, filtrado y etiquetado, de los minerales existentes en la base de datos que son de tipo granito, donde se consigue un etiquetado óptimo sobre las imágenes, dicho trabajo fue realizado por la universidad de Turquía en 2012 [12].

*Snehamoy Chatterjee* en 2012 [13] publica un artículo en *Applied Intelligence* donde se realiza la clasificación de rocas calizas mediante máquinas de soporte vectorial, consiguiendo clasificar el 96,2% de las rocas correctamente.

En 2013 en Polonia, se realiza la clasificación de 9 tipos de rocas, extrayendo las características del espacio del color *CIE Lab*, y aplicando el vecino más próximo para obtener la clasificación de las muestras, consiguiendo un porcentaje de éxito del 99,8 % [14].

Los últimos trabajos encontrados fueron de 2015 y 2016, donde se trabajó exclusivamente con tipos de rocas ígneas y en cada uno de ellos se utilizó métodos de clasificación distintos.

El primero de ellos fue realizado por *Hossein Izadi, JavadSadri, Nosrat-Agha Mehran*, [15] en Irán donde se realizó la clasificación mediante un algoritmo de agrupación por color, consiguiendo entre el 85,24 al 92,15 % de éxito en la clasificación.

El segundo de ellos también realizado en 2015 [16] consistió en extraer las características de las secciones finas de dos tipos de rocas ígneas, donde se empleó para la clasificación el esquema de votación mayoritaria, consiguiendo un porcentaje de acierto del 98 %.

Finalmente el último estudio encontrado y más reciente consiste en aplicar las redes neuronales, donde para la extracción de las características se realiza una segmentación y una clasificación por cascada, donde se extraen características texturales y de color, sobre secciones finas con luz polarizada y luz polarizada cruzada. Aplican la red neuronal en cascada analizando primero las características de color y después las de textura, consiguiendo un porcentaje de éxito del 93,81 % para un total de 6 tipos de rocas[17].

Como se observa desde hace al menos 40 años, la clasificación automática de rocas ha sido de gran interés, debido a que existen empresas dedicada a la extracción de materiales como oro, plata, carbón, etc. y estos estudios se han realizado principalmente por dicho motivo, ya que permite discriminar los tipos de rocas o materiales que no son de interés, por ejemplo en una mina de carbón, pudiendo separar los que son de un tipo y los que no.

En el Anexo A se encuentra la tabla resumen del estado del arte, donde se puede apreciar con más detenimiento lo descrito anteriormente.

### 1.3. OBJETIVOS DE LA PROPUESTA

Este TFM tiene por objetivo diseñar e implementar un sistema de identificación de rocas a partir de imágenes obtenidas desde dispositivo móvil. Para alcanzar este objetivo se van a desarrollar los siguientes objetivos secundarios:

- Crear una base de datos de imágenes de rocas capturadas desde un dispositivo móvil.
- Parametrizar las imágenes de la base de datos con la finalidad de destacar sus características más discriminantes.
- Aplicar algoritmos de aprendizaje supervisado que permita identificar los distintos tipos de rocas características de Gran Canaria, mediante imágenes sacada con dispositivos móviles.

#### 1.4. ESQUEMA DE LA MEMORIA

A continuación se describe como se plantea la realización del presente trabajo, los capítulos que lo compone y su contenido.

En un primer lugar se realiza una investigación sobre la identificación de rocas, trabajos realizados, algoritmos y aplicaciones creadas, con el fin de ver que se ha hecho y si se ha realizado algo similar al presente trabajo.

Una vez realizada dicha investigación se redacta el estado del arte descrito en el presente capítulo, para posteriormente realizar las salidas al campo con el fin de identificar las rocas a capturar para la posterior creación de la base de datos, como se explica en el capítulo II. Tras la creación de la base de datos, se estudia los distintos métodos mediante los cuales se puede realizar la extracción de características discriminatorias, así como las técnicas de aprendizaje supervisados a emplear para obtener la clasificación de las muestras, todo lo cual se describe en el capítulo III.

En el Capítulo IV, se explica cómo mediante el programa Matlab, se extraen las características, se clasifican las muestras por tipo, se aplica la clasificación por redes neuronales y se describen los resultados obtenidos, los cuales se comparan con el estado del arte.

Finalmente se redacta el Capítulo V, donde se describen las conclusiones, líneas futuras y las bibliografías empleadas para la realización del presente TFM.



# CAPÍTULO II: BASE DE DATOS



## 2.1. INTRODUCCIÓN

La identificación de rocas puede ser analizada de distintas formas, por su composición química, tamaño, texturas, etc. La diversidad de formas, tamaños, texturas, colores de las rocas hace difícil para un ojo no experimentado, saber de qué tipo se trata.

En nuestro caso, analizando la literatura existente para la identificación de rocas, se requiere de un conjunto de datos en la mayoría de caso imágenes, y en otros casos características como la composición química de cada tipo de roca a identificar, por ejemplo.

Por ello el primer paso para el desarrollo del presente TFM, consistirá en obtener una base de datos de los distintos tipos de rocas a identificar.

Tras varias reuniones con el tutor se determinó que se procedería a la obtención de las imágenes mediante la cámara de un *Smartphone*, obteniendo los distintos tipos de rocas en las cercanías del campus universitario, y excepcionalmente cuando se fuera algún lugar donde se pudieran obtener más tipos de rocas.

## 2.2. CONSTRUCCIÓN DE LA BASE DE DATOS

Para la construcción de la base de datos se determinó, que serían necesarios disponer de un mínimo de 50 imágenes por tipo de roca a identificar.

### 2.2.1. Toma de datos en campo.

Como se indicó anteriormente los tipos de rocas a identificar se encuentran en la zona del campus universitario. Debido a la falta de variedad de tipos de muestras se tomaron también imágenes de distintos elementos constructivos, como baldosas, hormigón visto o roca empleada para la construcción de las edificaciones y muros de las instalaciones del campus universitario.

Las tomas de los tipos de rocas, fueron realizadas en distintos días y horas, para obtener muestras en distintas condiciones de iluminación, con el fin de hacer más eficiente el sistema de identificación.

### 2.2.2. Localización de las tomas.

Las tomas se obtuvieron principalmente en el campus universitario, así como en La Isla de La Palma, aprovechando unas vacaciones, se tomando diversas muestras, en los distintos lugares visitados.

A continuación se describen los lugares donde se realizaron la toma por tipo de roca, enumeradas por orden alfabético.

- **Tipo A:** Roca de la Cantera de Arucas, usada en los edificios de telecomunicaciones para la base de las edificaciones y marcos de las puertas y ventanas.



Ilustración 2. 1. Ubicación de las muestras del tipo A

Ejemplo de muestra fotografiada



Ilustración 2. 2. Muestra del tipo de roca A

- **Tipo B:** Hormigón Visto, material usado en el edificio donde se encuentra la cafetería y zona de estudios.



Ilustración 2. 3. Ubicación de las muestras del tipo B

Ejemplo de muestra fotografiada:



Ilustración 2. 4. Muestra del tipo de roca B

- **Tipo C:** Rocas sueltas que se encuentran de manera decorativa en la rampa exterior de las escaleras de administración de teleco que dan a la cafetería.



Ilustración 2. 5 Ubicación de las muestras del tipo C



Ilustración 2. 6. Tipo de rocas a fotografiar del tipo C.

Ejemplo de muestra fotografiada



Ilustración 2. 7. Muestra del tipo de roca C

- **Tipo D:** Roca basáltica que se encuentra como maceteros de las palmeras a la entrada de la administración de telecomunicaciones y en parterre aledaño, esta roca es bastante empleada en el campus universitario, obteniéndose también muestras en el edificio de arquitectura. Este tipo de roca se encuentra en todas las zonas de la isla, al ser bastante empleado en decoración de exteriores.



Ilustración 2. 8 . Ubicación de las muestras del tipo D

Ejemplo de muestra fotografiada



Ilustración 2. 9. Muestra del tipo de roca D

A continuación se definen los siguientes tipos E, F y G, que por su cercanía describen conjuntamente, ubicándose las mismas en la zona de recreo / descanso de telecomunicaciones.

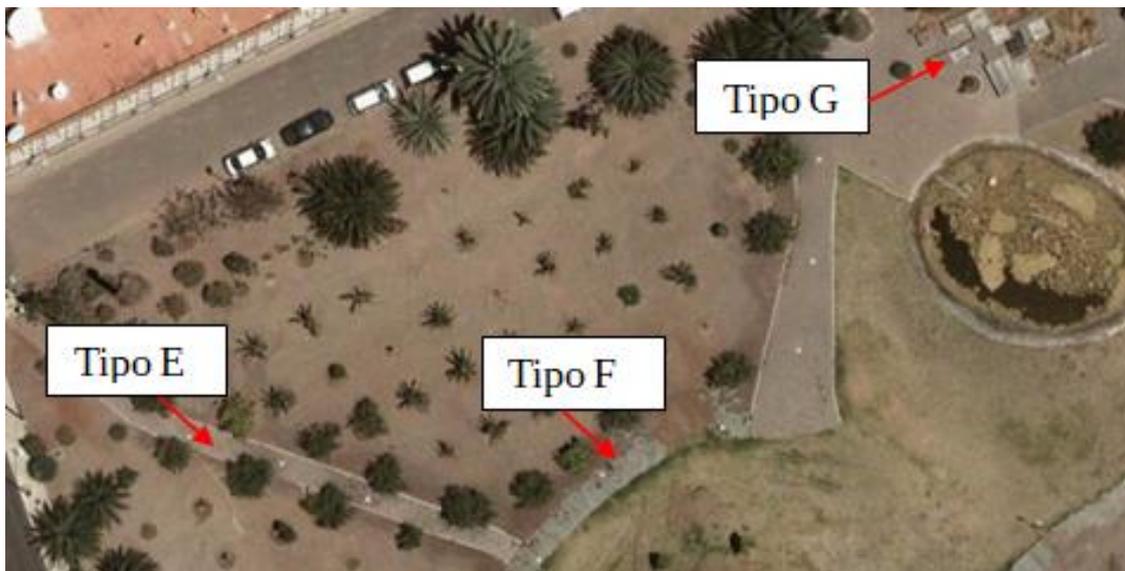


Ilustración 2. 10. Ubicación de las muestras E, F y G

- **Tipo E:** Baldosas de suelo donde se encuentran los limoneros que conducen al lago ubicado entre los edificios de teología y telecomunicaciones.



Ilustración 2. 11. Tipo de rocas a fotografiar del tipo E.

Ejemplo de muestra fotografiada



Ilustración 2. 12. Muestra de tipo de rocas del tipo E

- **Tipo F:** Baldosas del suelo que forman un camino y rodean al lago.



Ilustración 2. 13 Tipo de rocas a fotografiar del tipo F

Ejemplo de muestra fotografiada



Ilustración 2. 14. Muestra de tipo de rocas del tipo E

- **Tipo G:** Rocas que componen la estructura de las sillas y mesas de la zona de recreo o descanso donde se encuentra la iglesia, el edificio de telecomunicaciones y teología.



Ilustración 2. 15. . Tipo de rocas a fotografiar del tipo G.

Ejemplo de muestra fotografiada



Ilustración 2. 16. Muestra de tipo de rocas del tipo G

Las muestras H, I y J son baldosas de suelo que se encuentran en el módulo B del edificio de telecomunicaciones.

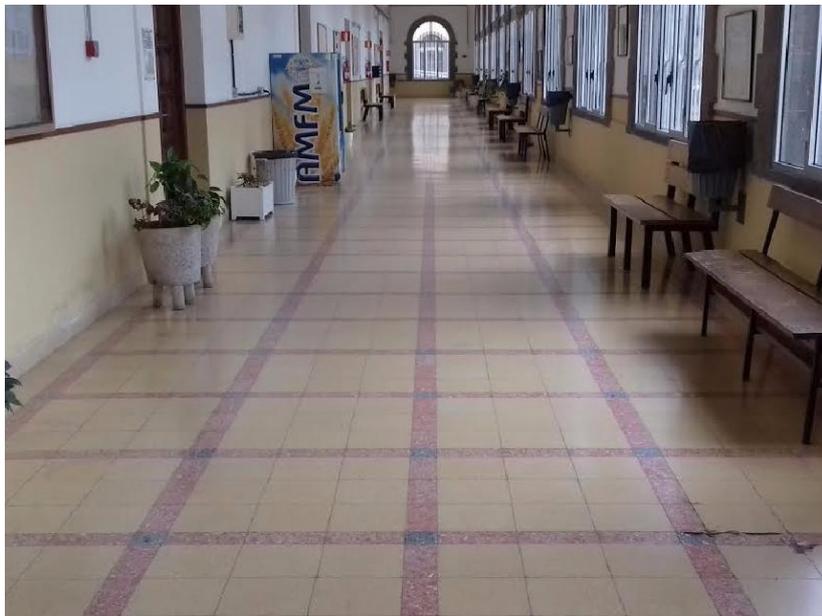


Ilustración 2. 17. . Tipo de rocas a fotografiar del tipo H, I, J.

- **Tipo H:** Línea de baldosas del edificio que divide en celdas de 4x3, el suelo a nivel decorativo

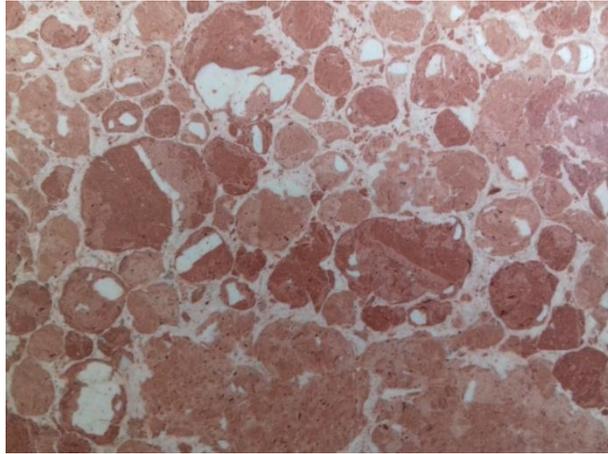


Ilustración 2. 18. Muestra de tipo de rocas del tipo H.

- **Tipo I:** Punto de unión entre las baldosas del tipo H.

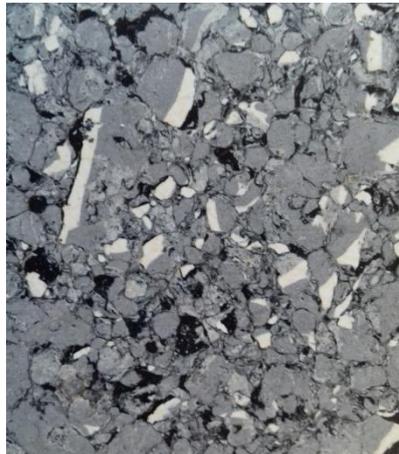


Ilustración 2. 19. Muestra de tipo de rocas del tipo I.

- **Tipo J:** Parte principal de la baldosa del suelo que junto con H e I forma el pasillo del edificio.

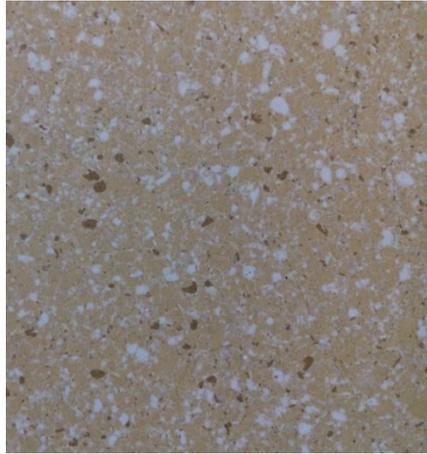


Ilustración 2. 20. Muestra de tipo de rocas del tipo J.

- **Tipo K:** Baldosas que forman un sendero en la zona de recreo, de distinto tipo a las anteriormente descritas.



Ilustración 2. 21. Ubicación de las muestras del tipo K



Ilustración 2. 22. Tipo de rocas a fotografiar del tipo K.

Ejemplo de muestra fotografiada



Ilustración 2. 23. Muestra de tipo de rocas del tipo K.

- **Tipo L:** Rocas porosa de color blanquecino/grisáceo que se encuentra en los aparcamientos del edificio de teología.



Ilustración 2. 24 .Ubicación de las muestras del tipo L



Ilustración 2. 25. Tipo de roca L.



Ilustración 2. 26. Muestra de tipo de rocas del tipo L.

Las muestras LL y M se corresponde con los muros existentes en el campus universitario (principalmente en teología y telecomunicaciones).

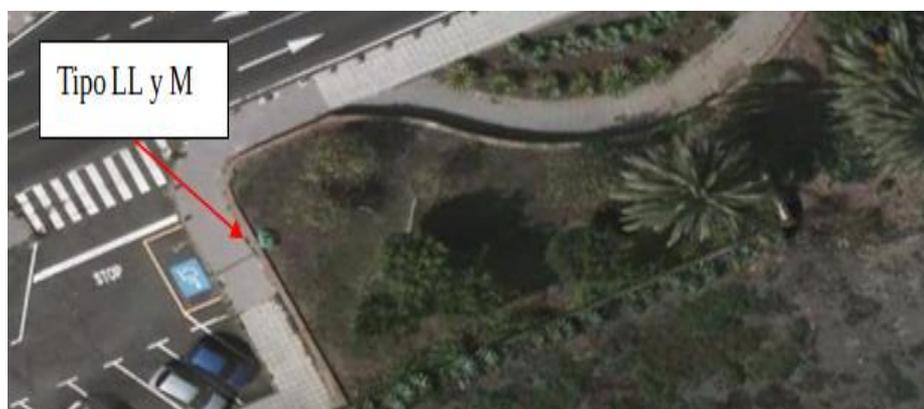


Ilustración 2. 27. Ubicación de las muestras LL y M



Ilustración 2. 28. Tipo de rocas a fotografiar del tipo LL y M.

- **Tipo LL:** Rocas de muros existentes en el campus universitario



Ilustración 2. 29. Muestra de tipo de rocas del tipo LL.

- **Tipo M:** Rocas que componen la parte de arriba del muro anterior.



Ilustración 2. 30. Muestra de tipo de rocas del tipo M

- **Tipo N:** Rocas existentes en la fachada de telecomunicaciones.



Ilustración 2. 31. Ubicación de las muestras del tipo N

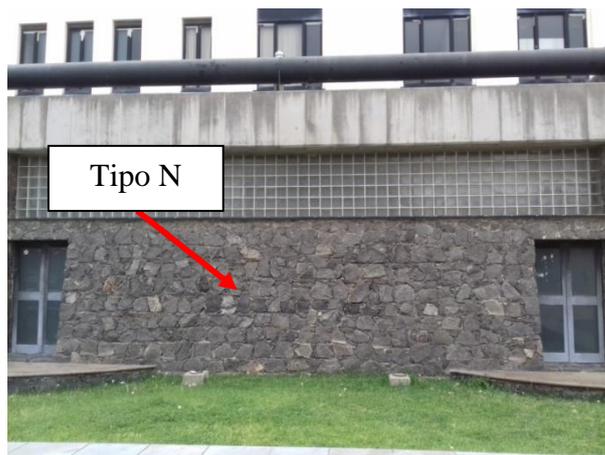


Ilustración 2. 32. Tipo de rocas a fotografiar del tipo N.

Ejemplo de muestra fotografiada



Ilustración 2. 33. Muestra de tipo de rocas del tipo N.

- **Tipo O:** Laja que forman parte de parterre, cercano a parada de guaguas en ingeniería y comedor universitario.



Ilustración 2. 34. Ubicación de las muestras del tipo O.



Ilustración 2. 35. . Tipo de rocas a fotografiar del tipo O.

Ejemplo de muestra fotografiada



Ilustración 2. 36. Muestra de tipo de rocas del tipo O.

Las siguientes muestras son las tomadas en la isla de La Palma.

- **Tipo P:** Roca que se encuentra bajando por el sendero, a la playa de Los Nogales, en Puntallana, La Palma.

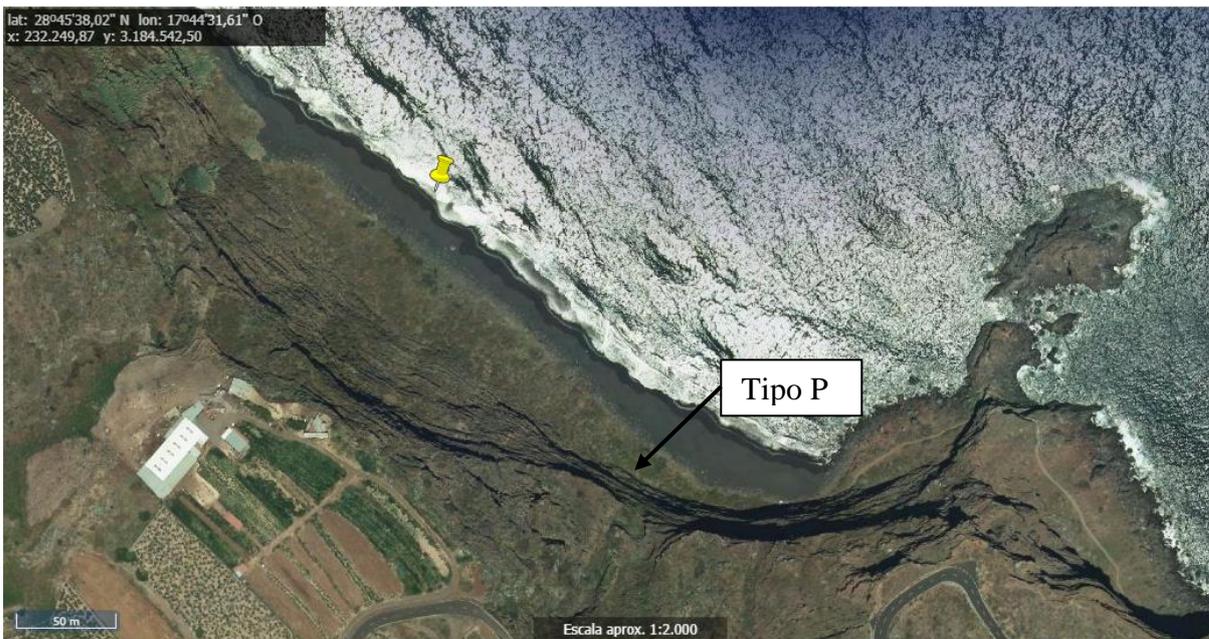


Ilustración 2. 37 Ubicación de las muestras del tipo P



Ilustración 2. 38. Tipo de rocas a fotografiar del tipo P.

Ejemplo de muestra fotografiada



Ilustración 2. 39. Muestra de tipo de rocas del tipo P.

- **Tipo Q:** Roca encontrada en el sendero de La Ruta de los Volcanes de Teneguía, en Fuencaliente, La Palma.



Ilustración 2. 40 Ubicación de las muestras del tipo Q.

Ejemplo de muestra fotografiada



Ilustración 2. 41. Tipo de rocas del tipo Q.

- **Tipo R:** Roca encontrada en El Roque de Los Muchachos, Garafía, La Palma.

Ejemplo de muestra fotografiada



Ilustración 2. 42. Muestra de tipo de rocas del tipo Q.

### 2.2.3. Características de la base de datos

Para la obtención de las imágenes se empleó un dispositivo móvil, Samsung Galaxy J36, cuya cámara de fotos tiene una resolución de 8 megapíxeles, siendo esta resolución suficiente para el objeto del presente trabajo.

Galaxy J3 Samsung	
	
<b>Cámara</b>	
<b>Cámara principal</b>	
<b>Resolución</b>	8 megapíxeles
<b>Flash</b>	LED
<b>Vídeo</b>	Full HD 1080p
<b>Cámara frontal</b>	
<b>Resolución</b>	5 megapíxeles
<b>Flash</b>	No definido

**Ilustración 2. 43. Características técnicas de la cámara del sensor de captura de imágenes.**

Se identificaron un total de 19 tipos de rocas, obteniendo 50 muestras por tipo, componiéndose nuestra base de datos de un total de 950 imágenes.

Siendo las características principales de las imágenes las siguientes:

FORMATO COMPRESIÓN	JPEG
DIMENSIONES	2448 x 3264
ANCHO	12,76 cm
ALTO	17 cm
RESOLUCIÓN HORIZONTAL	96 ppp
RESOLUCIÓN VERTICAL	96 ppp
PROFUNDIDAD DEL BIT	8

**Tabla 2. 1 Características técnicas de la cámara del sensor de captura de imágenes.**

Tras tener el número total de muestra se determinó que era necesario clasificar las mismas por su tipo y a su vez si se emplearían para entrenar o para testear el sistema de clasificación empleado.

Se determinó que para el entrenamiento se emplearía el 70 % (35 muestras) y el 30 % (15 muestras) para el testeo, por tipo de roca.

Para ello se ordenó las imágenes modificándoles el nombre y usando un código que permita la perfecta identificación de clase y tipo de cada muestra, pudiendo trabajar y procesar las imágenes en Matlab, siendo este código descrito como:

- C: Indica que se trata de una imagen de entrenamiento
- Ct: Indica que se trata de una imagen de testeo
- Primer valor numérico: Indica a qué tipo de roca pertenece
- Guión bajo: separa el tipo del número de muestras
- Segundo valor numérico: Indica el número de muestra que se trata.

Siendo un ejemplo de lo anterior, el que a continuación se describe:

1. Tipo de roca, C
2. Para entrenamiento
3. Última muestra

Se obtiene la imagen con el siguiente código: **C3\_35.jpg**

# CAPÍTULO III: MÉTODOS APLICADOS



### 3.1. INTRODUCCIÓN

Los distintos experimentos encontrados en las bibliografías existentes, tenían un conjunto de procedimientos comunes, en primer lugar se obtienen las imágenes, a continuación pre procesan las imágenes, posteriormente se le aplica un método de extracción de características y finalmente se emplea un algoritmo de clasificación.

Por nuestra parte el pre procesado no se aplica debido a la falta de tiempo para la realización del trabajo, centrándose el mismo, en los métodos de extracción y clasificación, con el fin de analizar los datos obtenidos en función del método de extracción y algoritmos de clasificación empleados, los cuales se explican en el presente capítulo.

### 3.2. MÉTODOS DE EXTRACCIÓN DE CARACTERÍSTICAS

Los métodos a emplear para la extracción de características de las imágenes son dos, el *Local Binary Pattern* y el *Discrete Wavelet Transform*, en adelante, LBP y DWT, respectivamente. Estos métodos permiten la extracción de características indistintamente del color, la rotación y tamaño de las imágenes, siendo estos los motivos por los cuales se eligen estos.

#### 3.2.1. LOCAL BINARY PATTERN (LBP)

Se trata de una técnica descriptiva pero potente [18], que nos permite clasificar la imagen por valores binarios representativos.

Este método trabaja en un único canal, escalas de grises, rojo, verde o azul, donde se analiza pixel por pixel la imagen. Cada pixel se compara con los que lo rodean, si el valor del pixel a analizar, es igual o mayor al valor del pixel vecino, se le asigna el valor 1, en caso contrario el valor es 0.

$$\begin{aligned} \text{Valor del vecino} \geq \text{Valor central} &\Rightarrow 1 \\ \text{Caso contrario} &\Rightarrow 0 \end{aligned} \tag{3.1}$$

Cuando los pixeles vecinos son convertidos a 0 y 1, se pasa a formato decimal, obteniendo el valor del pixel central.

Este método permite la resolución de múltiples problemas debido a su capacidad discriminadora.

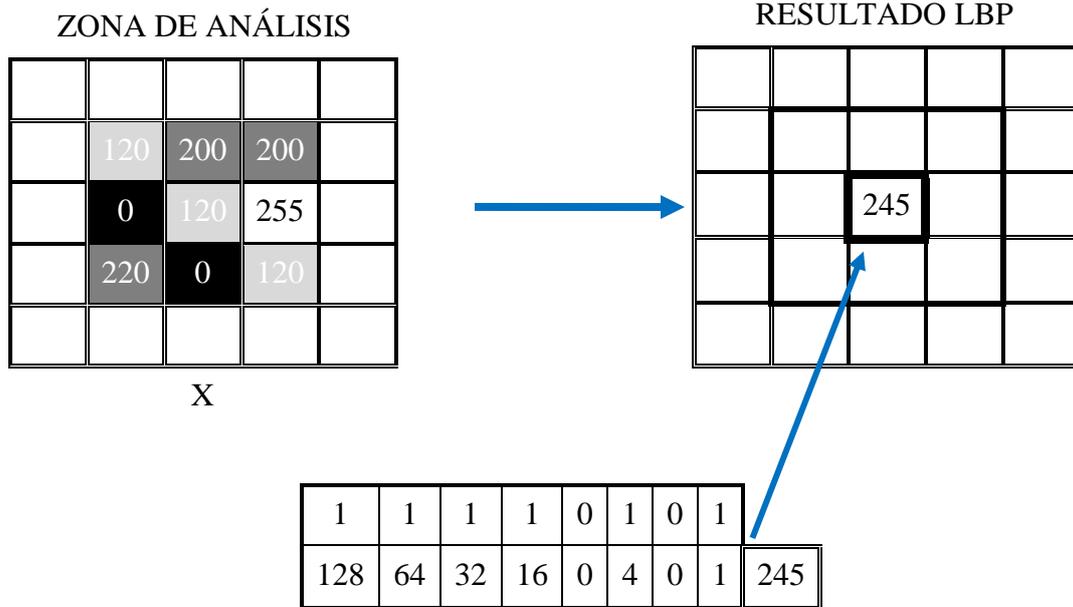


Ilustración 3. 1. Cálculo del LBP.

Para los bordes de la imagen, la literatura indica la estrategia a seguir, siendo las dos opciones más usadas o comunes, las siguientes:

- Dar valores iguales a los colindantes, a los pixeles sin valor.
- No calcular el *LBP* de la zona, siguiendo el mismo criterio para todo el proceso de cálculo de los bordes.

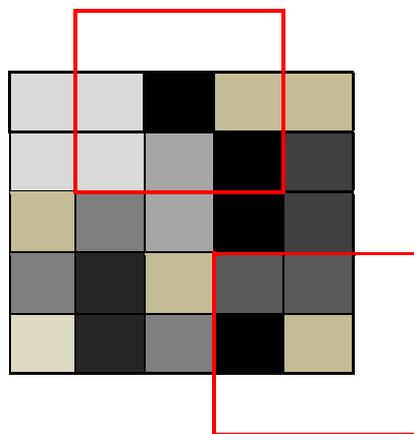


Ilustración 3. 2. Análisis de los bordes, LBP.

Finalmente se obtiene un vector, donde se encuentra los datos de *LBP* de cada pixel de la imagen, obteniendo el denominado descriptor de la imagen, el cual representa el verdadero valor representativo de la misma.

Posteriormente es posible analizar este vector y obtener el histograma *LBP*, como se puede apreciar en la siguiente ilustración.

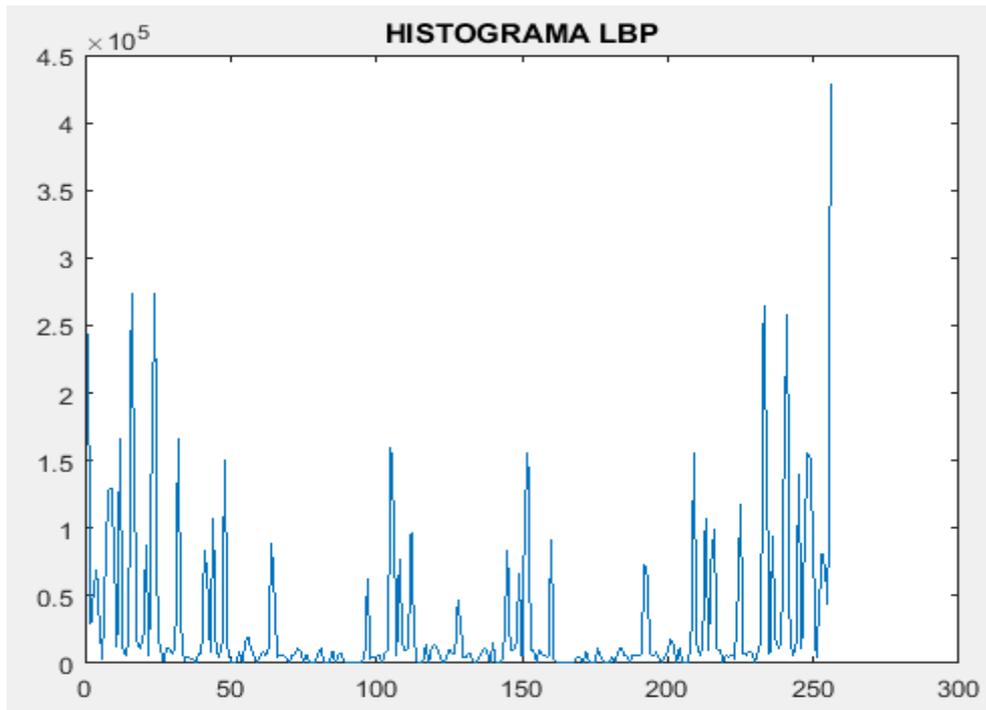


Ilustración 3. 3. Histograma del LBP.

Se suele definir un umbral que permite mejorar el *LBP*, en donde se realiza una distinción dado un umbral determinado, si al restar el pixel vecino con el pixel central este es mayor que el umbral, se le asigna el valor 1, en el caso contrario 0.

$$\begin{aligned} \text{Valor del vecino} - \text{Valor central} \geq \text{Umbral} &\Rightarrow 1 \\ \text{Caso contrario} &\Rightarrow 0 \end{aligned} \quad (3.2)$$

### 3.2.2. DISCRETE WAVELET TRANSFORM (DWT)

La transformada de *Wavelet* es un método que permite la compresión de audio (1D) o de imágenes (2D). En el cual se analizan un grupos de funciones básicas producidas por el desplazamiento y el escalado de una función madre o principal, a las cuales se les realiza un proceso de análisis de la *Wavelet*, para su representación mediante dichas funciones.

En este caso el proyecto se centra en analizar la transformada para las dos dimensiones, ya que se trabaja con imágenes.

#### 3.2.2.1. DWT 2D

La *DWT* bidimensional [19] se emplea para la compresión y análisis de multi-resolución en la ciencia de la inteligencia artificial. Para computar las imágenes se emplean filtros 2D calculados a partir de la convolución de filtro de 1D sobre los mismos. Los cuales pueden ser separables o no, siendo separable cuando:

$$f(n_1, n_2) = f_1(n_1) * f_2(n_2) \quad (3.3)$$

Si es separable se descompone una imagen  $S_i(n_1, n_2)$  en una imagen promedio y en tres imágenes de detalle de acuerdo a las expresiones siguientes:

$$\begin{aligned}
 S_{i+1}(n_1, n_2) &= \sum_{k_1} \sum_{k_2} g(k_1)g(k_2)S_i(2n_1 - k_1, 2n_2 - k_2) \\
 W_{i+1}^1(n_1, n_2) &= \sum_{k_1} \sum_{k_2} g(k_1)h(k_2)S_i(2n_1 - k_1, 2n_2 - k_2) \\
 W_{i+1}^2(n_1, n_2) &= \sum_{k_1} \sum_{k_2} h(k_1)g(k_2)S_i(2n_1 - k_1, 2n_2 - k_2) \\
 W_{i+1}^3(n_1, n_2) &= \sum_{k_1} \sum_{k_2} h(k_1)h(k_2)S_i(2n_1 - k_1, 2n_2 - k_2)
 \end{aligned}
 \tag{3.4}$$

Donde  $H(z)$  y  $G(z)$  son los filtros *Wavelet* 1D.

La señal  $S_{i+1}(n_1, n_2)$  es un suavizado de baja resolución, empleando un filtro de paso bajo y diezmado por 2 a lo largo de filas y columnas.

Las señales  $W_{i+1}^1(n_1, n_2)$ ,  $W_{i+1}^2(n_1, n_2)$  y  $W_{i+1}^3(n_1, n_2)$  contiene el detalle de  $S_{i+1}(n_1, n_2)$ .

El primer nivel de la DWT 2D, se calcula como se muestra en la siguiente imagen.

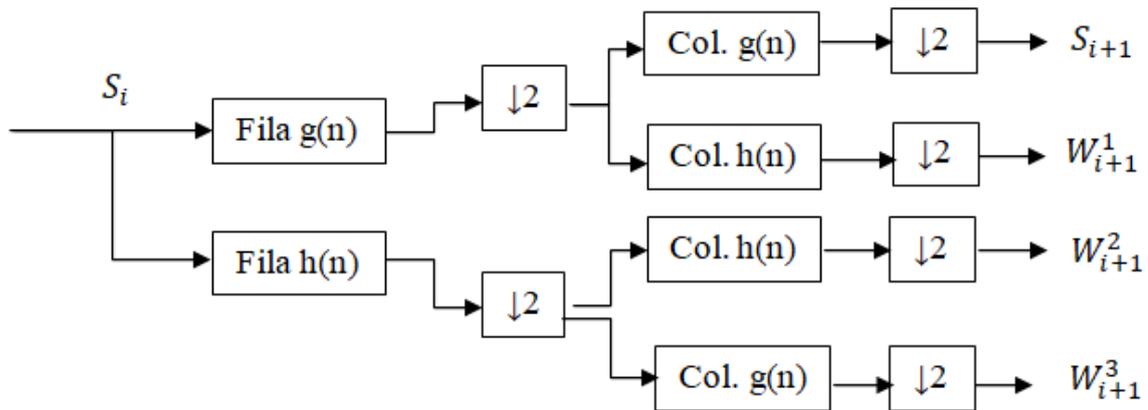


Ilustración 3. 4. Diagrama DWT.

Al tratar con imágenes linealmente no separables, esta se descompone en cuatro sub-imágenes, resultado de aplicar *DWT* por filas y columnas. En la siguiente imagen se puede ver la descomposición para un primer nivel.

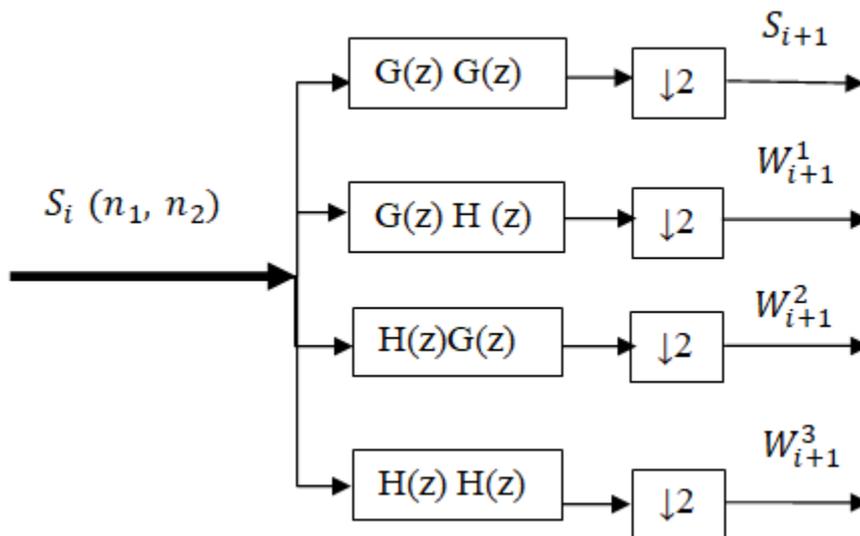


Ilustración 3. 5. Diagrama DWT linealmente no separable.

Se debe tener en cuenta que normalmente las imágenes no son separables linealmente, siendo necesario siempre aplicar el *DWT* no separable.

Dentro del *DWT* se dispone de diversas arquitecturas, siendo las principales:

- Aproximación directa
- Par-Impar Paralela
- Recurrente

En este trabajo se usa la arquitectura de aproximación directa.

### 3.2.2.2. Arquitectura de aproximación directa.

Consiste en usar un algoritmo piramidal, aplicándole a la imagen 2D, 4 filtros modulados siendo estos los siguientes:

- GG representa el filtro de paso bajo-bajo (la imagen original suavizada por el filtrado).
- GH representa el filtro de paso bajo-alto (las variaciones en la componente horizontal de la imagen).
- HG representa el filtro de paso alto-bajo (las variaciones en la componente vertical de la imagen).
- HH representa el filtro de paso alto-alto (las variaciones de la imagen que no se pueden obtener por los filtros anteriores, GH y HG, donde se muestra las variaciones de la componente de la diagonal de la imagen).

En la siguiente imagen se ve la arquitectura descrita con cuatro niveles de *DWT*:

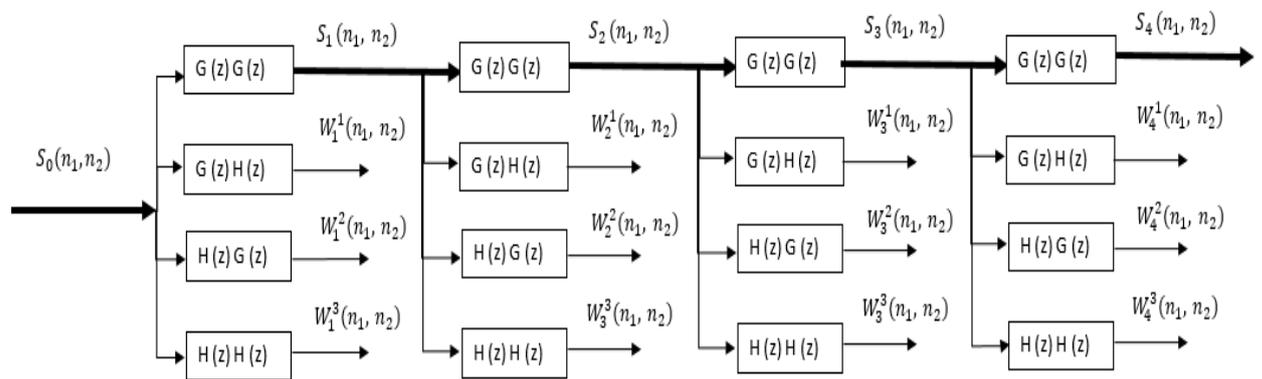


Ilustración 3. 6. Esquema de DWT no separable, por arquitectura por aproximación directa, cuarto nivel.

Como se muestra en la imagen anterior el DWT, se aplica sobre el filtro GG, ya que es la imagen con más información.

En el siguiente ejemplo se observa lo anteriormente dicho, donde en la ilustración (a), se muestra el esquema de la arquitectura de aproximación directa de las distintas imágenes hasta llegar al cuarto nivel.

Y en (b) se muestra una imagen de la base de dato, en el cuarto nivel DWT, al aplicar la inversa de dicha transformada, donde se obtiene las imágenes GG4, GH4, HG4 y HH4.

(a)

GG4	GH4	GH3	GH2	GH
HG4	HH4			
HG3		HH3		
HG2		HH2		
HG			HH	

(b)

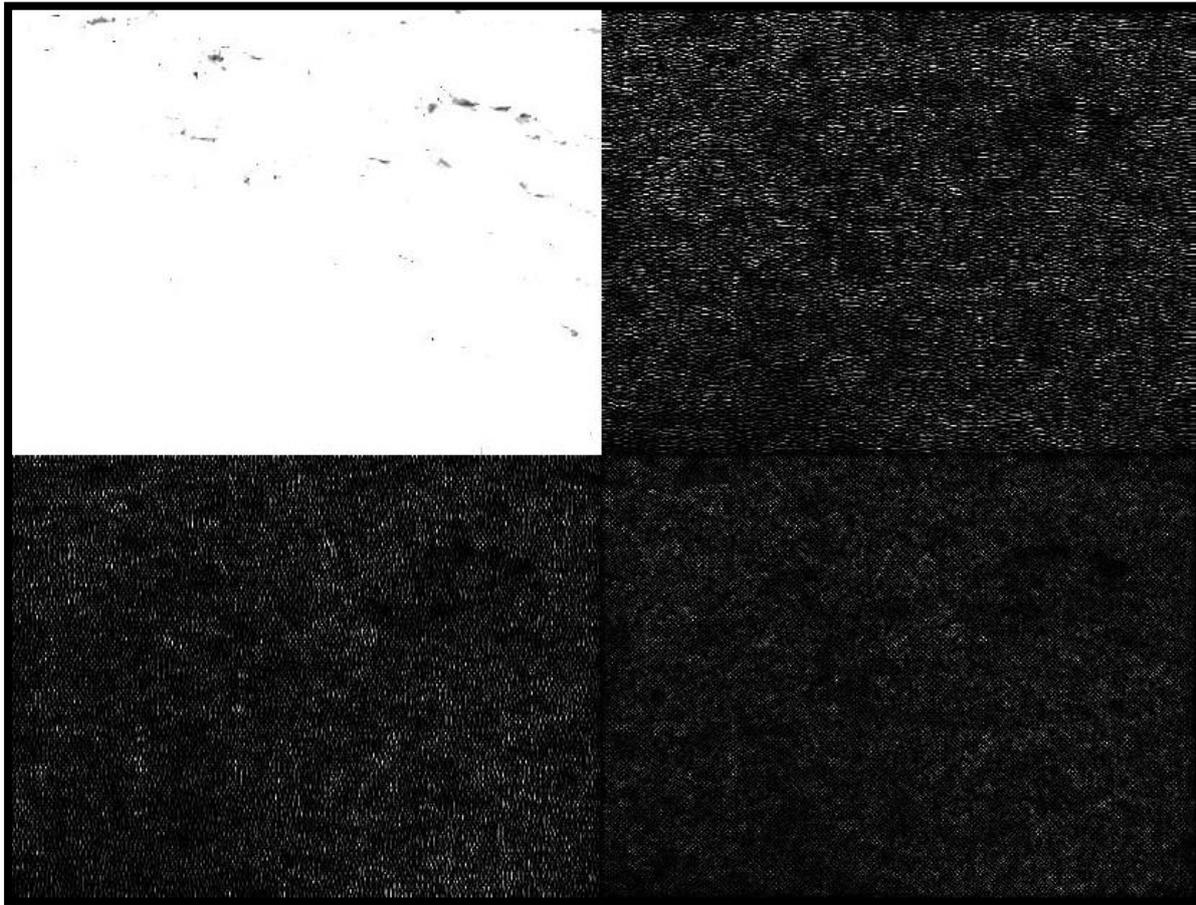


Ilustración 3. 7. Esquema de DWT de una imagen por aproximación directa.

### 3.3. REDES NEURONALES ARTIFICIALES O ARTIFICIAL NEURONAL NETWORK (ANN)

Se trata de una técnica empleada en el aprendizaje automático (*machine learning*), que pretende imitar las redes neuronales de los seres vivos.

En los años 40 y 50 se desarrollan las primeras teorías sobre las redes neuronales, pero no es hasta 1960, cuando *Roseblatt Widrow y Hoff*, crean la primera red denominada *ADELIN*, para uso industrial [20].

El funcionamiento de las redes neuronales biológicas se basa en impulsos eléctricos, dichas neuronas están conectadas entre sí y trabajan de manera conjunta, adquiriendo experiencia y aprendiendo, reforzando las conexiones entre ellas.

Compuestas por:

- Cuerpo (*SOMA*): Transforma entradas en salida, rodea al núcleo.
- Dendritas: fibras que reciben las actividades
- Axón: Fibras que conecta las neuronas entre sí.

En la siguiente imagen se ve las partes descritas:

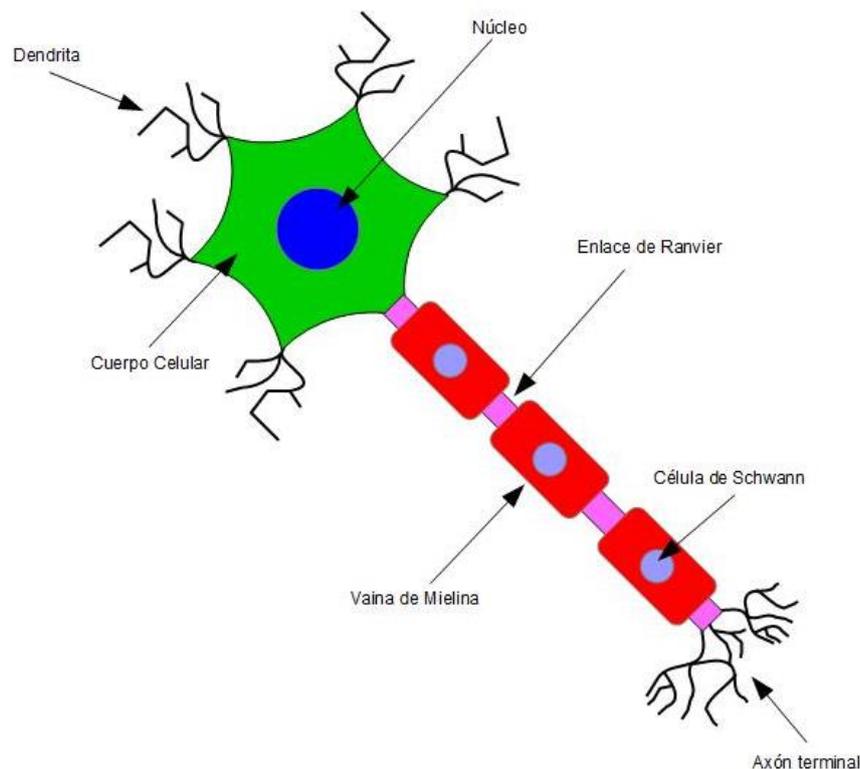


Ilustración 3. 8. Esquema de una Neurona

Entre las dendritas y el axón se produce un enlace químico denominado sinapsis. Las señales entran por las dendritas hacia el cuerpo, y se dispara un pulso energético cuando acumula energía suficiente, que se envían a otras neuronas mediante el axón.

Las redes neuronales artificiales intentan seguir este principio básico, para ello se ha centrado en el desarrollo matemático y estadístico.

Estas redes artificiales funcionan de la siguiente forma, se usan unos datos de entrada, que hay que combinar para predecir un cierto resultado, datos de salida.

Para poder predecir los resultados es necesario el entrenamiento de la red, que consiste en ajustar lo mejor posible la combinación de parámetros más adecuada, para obtener un resultado de salida.

Una vez entrenada, se puede clasificar o predecir, las salidas con un cierto éxito en función de los parámetros combinados.

Dentro de las redes neuronales artificiales se encuentran múltiples modelos, los cuales se describen a continuación [21]:

- **Perceptrón**
- Adaline
- **Perceptrón multicapa**
- Memorias asociativas
- Máquina de *Boltzmann*
- Máquina de *Cauchy*
- Propagación hacia atrás (*backpropagation*)
- Redes de Elman
- Redes de Hopfield
- Red de contrapropagación
- Redes de neuronas de base radial
- Redes de neuronas de aprendizaje competitivo
- Mapas Autoorganizados (RNA) (Redes de *Kohonen*)
- Crecimiento dinámico de células
- Gas Neuronal Creciente
- Redes ART (*Adaptative Resonance Theory*)

Así mismo son clasificables de las siguientes formas:

Por topología:

- Redes de propagación hacia delante o a cíclicas, no existen ciclos ni conexiones entre neuronas de la misma capa, las señales van desde la capas de entrada hacia la de salida.
  - Mono capas, como puede ser la del perceptrón o Adaline
  - Multicapas, por ejemplo perceptrón multicapa
- Las redes recurrentes, realizan al menos un ciclo cerrado para la activación neuronal, como son las redes *de Elman*, *Hopfield* y máquina de *Botzmann*.

Aprendizaje:

- Supervisado: consiste en que se tendrá un conjunto de datos de entrada previamente clasificados y conocidos.
- No supervisado: No se emplea un conjunto de entrada de datos conocido.
- Redes híbridas: es una combinación de las dos anteriores para mejorar la convergencia.
- Aprendizaje reforzado: Combina el aprendizaje supervisado con el auto organizado.

Tipo de entrada:

- Redes analógicas: Procesa datos de entrada continuos y acotados
- Redes discretas: Procesan datos de entrada discretos y normalmente booleanos.

### 3.3.1. Red neuronal Perceptrón

El perceptrón es la unidad elemental de una red neuronal, la capa perceptrón dispone de varios datos de entradas, los cuales tienen asociados un peso y en el caso de superar un umbral dado, la salida será uno o por contra cero.

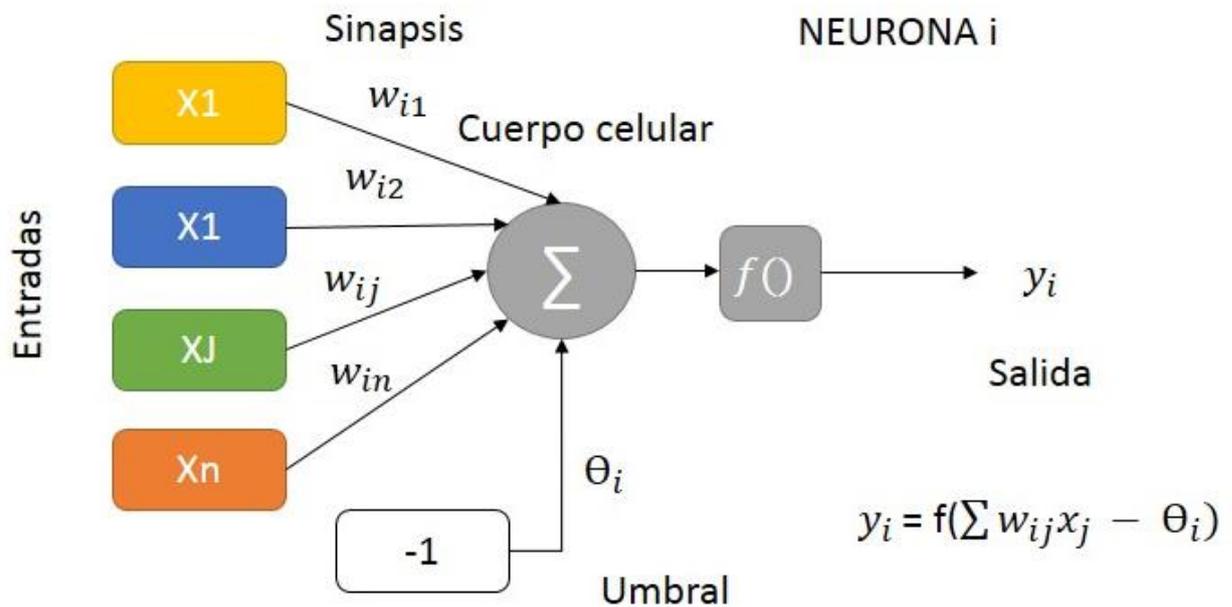


Ilustración 3. 9. Esquema de una ANN

Como se puede ver en la imagen anterior, hay una serie de datos de entrada que pasan por las dendritas, las cuales llegan al cuerpo de la célula y en caso de superar el umbral, acumulación de energía o *spike*, se obtiene unos datos de salida, por el axón.

### 3.3.2. Red neuronal Perceptrón Multicapa

Si bien la red anterior funciona con múltiples situaciones sencillas, cuando la situación se complica (problemas linealmente no separables), esta red no resuelve los problemas de dicha índole.

Las redes neuronales están conectadas entre sí, debido a las capas intermedias u ocultas, que nos permita realizar una discriminación de los datos, para mejorarlos, adaptarlos y obtener los resultados de salida deseados o esperados[20].

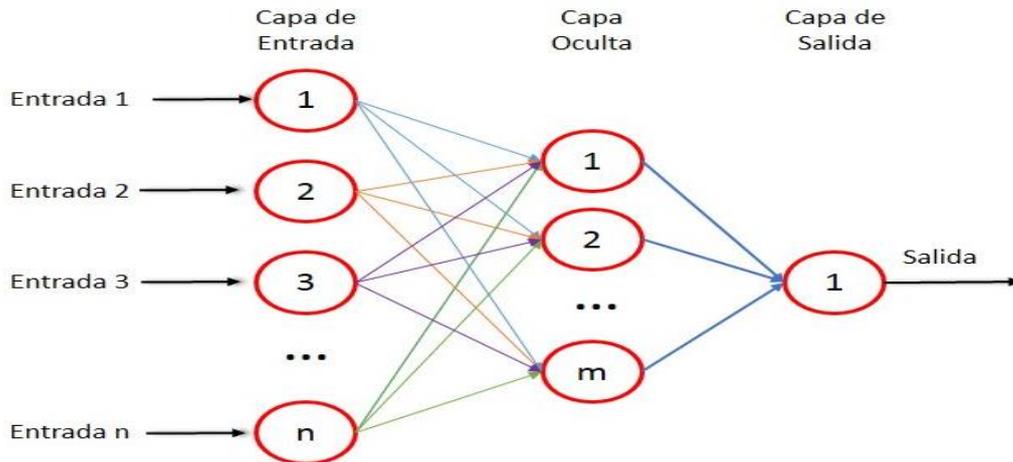


Ilustración 3. 10. Esquema de una ANN

La capa oculta se ve afectada por cada neurona de entrada, en caso de tener más capas ocultas, las neuronas se verían afectadas por las capas ocultas, hasta obtener el dato de salida, denominándose este procedimiento como algoritmo de retro propagación.

Las principales ventajas de este método son:

- Aprendizaje adaptativo
- Tolerancia a fallos
- Operaciones en tiempo real

Los cálculos se realizan en paralelo, por lo cual es necesario un equipo que permita ejecutar las redes de manera efectiva, ya que si el equipo empleado no dispone de unos buenos procesadores, no será capaz de llevar a cabo la ejecución de la red, pues consume una gran cantidad de recursos.

Así mismo la red neuronal no tiene unas reglas definitorias que sirvan para construir la red, número de neuronas por capas, arquitectura, número de capas, etc., sino que estas se definen de manera aleatoria en la asignación de pesos.

# CAPÍTULO IV: METODOLOGIA EXPERIMENTAL Y RESULTADOS



#### 4.1. INTRODUCCIÓN

En capítulos anteriores se explica el estado del arte en cuanto a clasificación de rocas mediante análisis de imágenes, cómo se obtienen las imágenes para la base de datos y las técnicas a emplear para clasificarlas. En el presente capítulo se describe la metodología seguida así como los resultados obtenidos, para finalizar se compara con el estado del arte.

#### 4.2. METODOLOGÍA

Tras la obtención de los datos, el siguiente paso es trabajar con ellos. Para ello se realiza una serie de experimentos, donde se emplean las técnicas descritas en capítulos anteriores, siendo un total de tres experimentos:

1. Clasificación por *LBP*.
2. Clasificación por *LBP* de los filtros de paso bajo-bajo (GG) del *DWT*, mediante arquitectura de aproximación directa hasta el cuarto nivel, de cada una de las sub-imágenes.
3. Clasificación por *LBP* de los filtros de paso bajo-bajo (GG) del *DWT*, mediante arquitectura de aproximación directa hasta el cuarto nivel, del conjunto de sub-imágenes.

De manera esquemática se observa a continuación los experimentos aplicados sobre la base de datos:

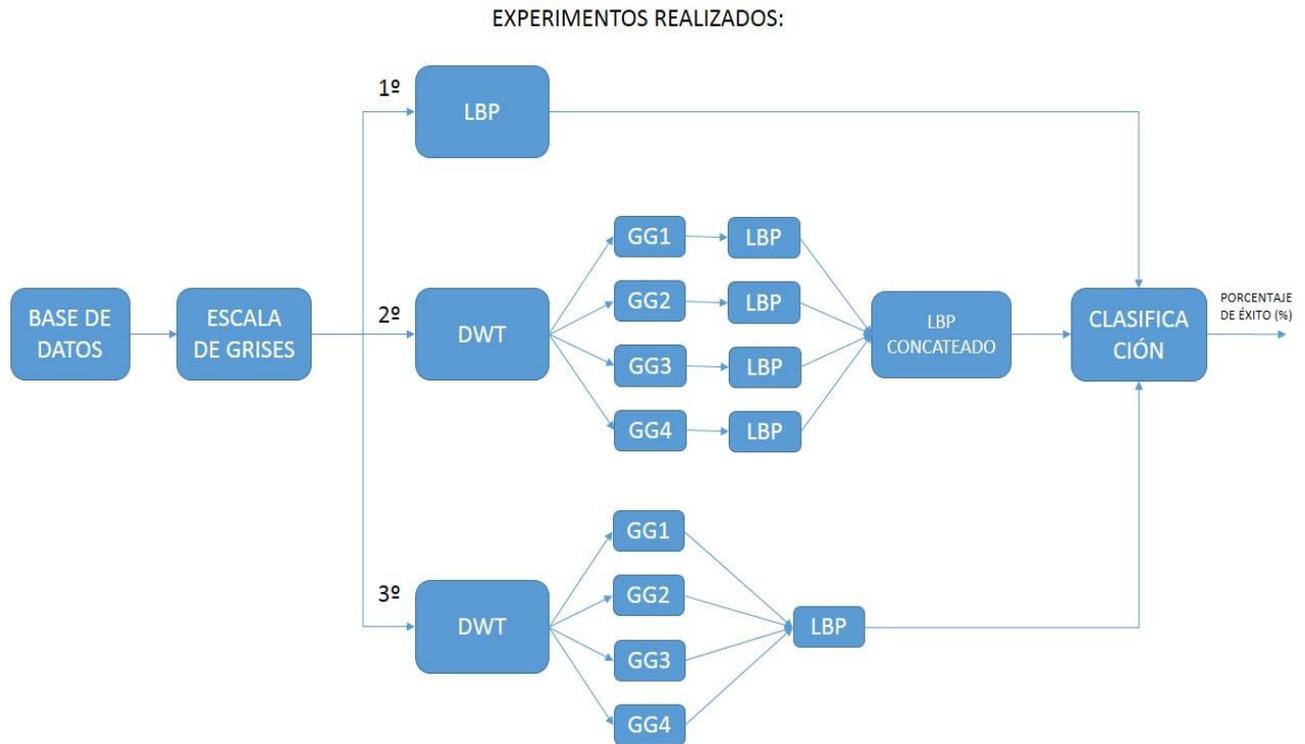


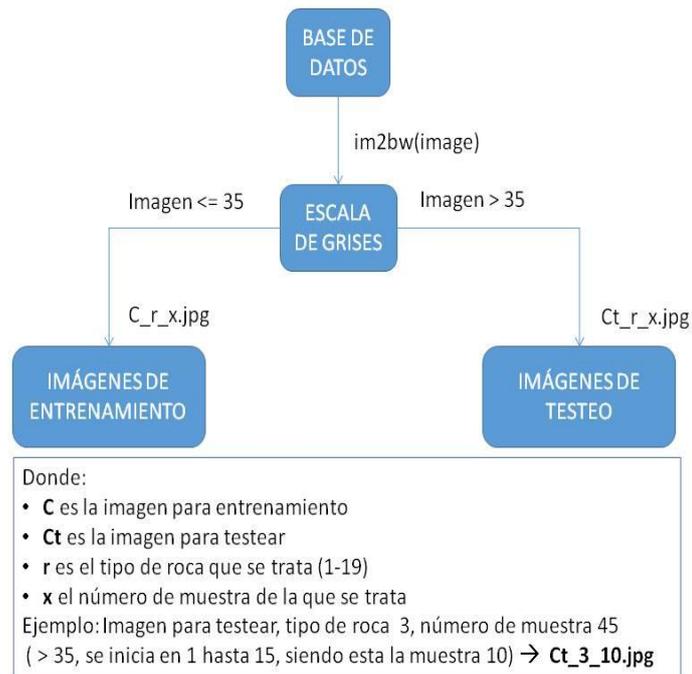
Ilustración 3. 11. Esquema de los experimentos realizados.

En los siguientes apartados se explicará de manera detallada cada uno de los pasos del esquema anterior.

#### 4.2.1. Base de datos a escala de grises.

Como se explica en el capítulo III, para obtener el *LBP* de una imagen, esta ha de estar en un único canal, para ello se transforma la imagen a escala de grises mediante *Matlab*, donde de manera automática nos convertirá la base de datos a escala de grises, mediante el comando *rgb2gray ()*, para posteriormente dar nombre a cada una de las imágenes en función de si se emplean para entrenamiento o testeo. Se sigue como criterio que las 35 primeras muestras son para entrenamiento y las restante para testear, empleando para ello el condicional *if*. Mediante un bucle *for* se recorren todas las imágenes de la base de datos y se transforman a escala de grises, obteniendo así la base de datos deseada.

En el siguiente diagrama se puede ver el funcionamiento del algoritmo descrito:



**Ilustración 4. 1. Esquema conversión a escala de grises.**

#### 4.2.2. Primer experimento: Clasificación por LBP.

Con las imágenes convertidas a escala de grises, el siguiente paso es conseguir las características de interés, donde el método empleado es el *LBP*, con este se obtiene un vector de una fila por doscientos cincuenta y seis columnas.

Donde cada columna se corresponde con el valor del *LBP* por pixel de la imagen, en la siguiente imagen se puede ver el histograma de frecuencia, calculado para una de las imágenes de la base de datos en escala de grises. En el mismo se puede apreciar en el eje *x* cómo se va desde 0 hasta 255 valores, siendo cada uno de ellos los pixeles que componen la imagen, y en el eje *y* el valor *LBP* que le corresponde.

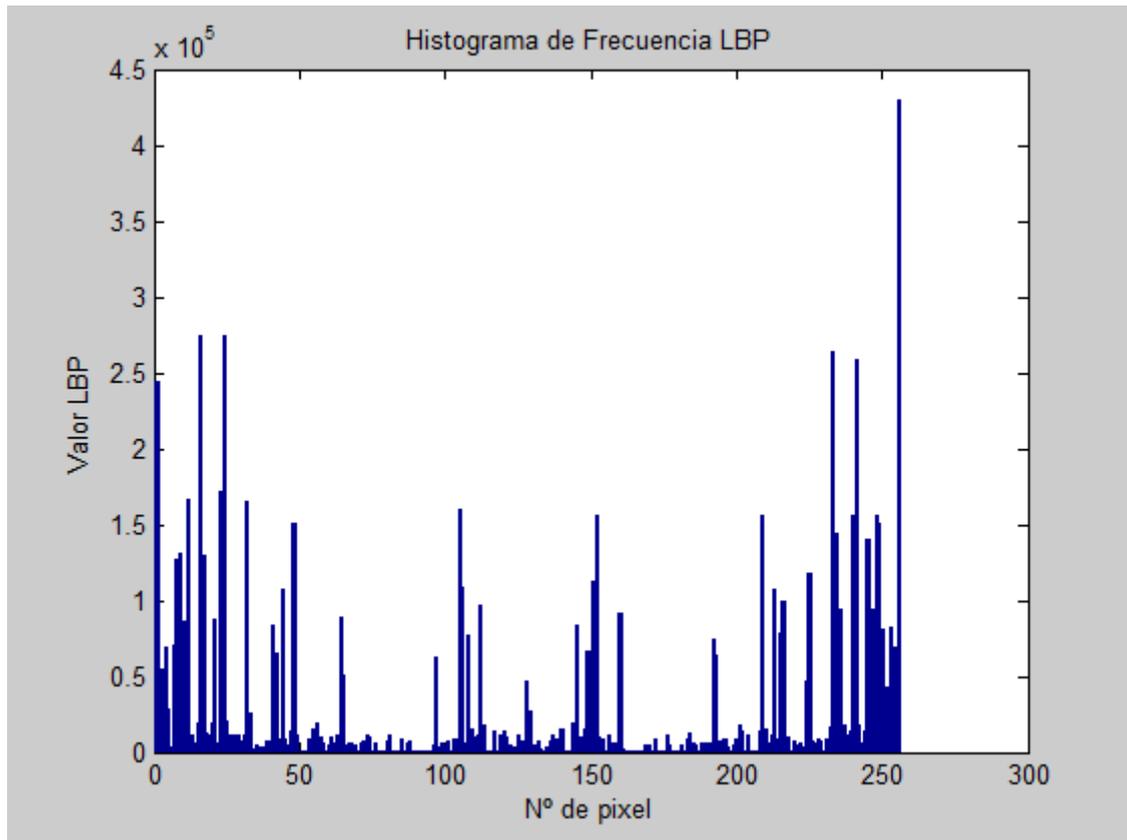


Ilustración 4. 2. Histograma de frecuencia LBP.

A continuación se crean las matrices que aglutinan el LBP de las imágenes de entrenamiento y de testeo, siendo estas las matrices de parámetros de la red neuronal, denominadas respectivamente como *param* (256x665) y *param2* (256x285).

Posteriormente se crea una matriz indicando que datos se corresponde con que muestra, siendo estas las matrices de etiquetas de la red, denominadas *target* (19x665) y *target2* (19x285).

Con estas etiquetas se puede relacionar los parámetros de entrada con las muestras, donde por ejemplo si se tienen 3 muestras de 3 tipos distintos de rocas, se puede saber perfectamente las muestras que pertenecen a un tipo de roca determinado.

	M1	M2	M3	M4	M5	M6	M7	M8	M9
A	1	1	1	0	0	0	0	0	0
B	0	0	0	1	1	1	0	0	0
C	0	0	0	0	0	0	1	1	1

Tabla 4. 1. Matriz de etiqueta, ejemplo 3 muestra por 3 tipos de rocas.

Como se puede observar en la tabla anterior, se tienen que las muestras del tipo A, son (M1, M2 y M3), ya que su valor es igual a uno, siendo el resto de muestras de los tipos B y C.

Tras esto se inicializa la red, la cual se entrena con los datos de entrenamiento y se obtienen unos resultados de salida.

Estos resultados de salida se deben aproximar al valor de la etiqueta de entrada, es decir, si se tiene que para la matriz de etiqueta (*target2*), el primer valor es 1, la etiqueta de salida (*etiquetasalida*) para esa posición en la matriz, debe ser muy cercano a este valor o igual, en caso contrario la red no clasifica bien esta muestra.

	15	16	17	18	19	20
1	0.4574	0.0208	0.0541	-0.0182	0.0080	-0.0329
2	-0.0184	0.9364	0.9723	0.8211	0.9007	0.7808
3	0.0838	0.0623	0.1079	0.0436	0.0161	0.0577
4	0.0014	-0.0501	-0.0059	-0.0441	-0.0469	0.0750
5	0.2299	0.0700	0.0983	0.0444	0.0520	-0.0702
6	-0.0036	-0.0799	-0.0556	-0.0479	-0.0388	0.1454
7	-0.1085	0.0833	0.1436	0.0867	-0.0040	0.0301
8	0.0100	-0.0275	0.1029	0.1389	0.2114	0.1078
9	-0.0697	-0.0785	-0.0605	-0.1440	-0.1038	-0.1101
10	0.0462	0.1590	0.0625	0.2432	-0.0181	0.0394
11	0.1031	0.1063	0.0219	0.0731	0.0745	0.1007
12	-0.0681	-0.1503	-0.1209	-0.0261	-0.0380	0.0948
13	0.1797	0.1006	-0.0102	-0.0063	0.1209	-0.0052
14	0.1232	0.0452	-0.0616	0.0035	0.0108	-0.0336
15	0.0768	-0.0308	-0.0376	0.0295	0.0569	-0.0179
16	0.0265	-0.0491	0.0390	0.0717	0.0302	0.0272
17	-0.0695	0.0382	-0.0378	-0.0154	-0.0511	-0.0027
18	0.0440	0.0306	0.0458	-7.1572e-04	0.0321	-0.0160
19	0.0073	-0.0387	-0.0227	-0.0221	-0.0968	0.0435

Ilustración 4. 3. Extracto de etiquetas de salida, roca tipo B, simulación con 24 neuronas una única iteración, análisis de tipo roca B.

Como se puede observar en la imagen anterior, se ve que el tipo de roca B definido en el capítulo II, se clasifica bastante bien, ya que los resultados se acercan bastante a 1. Como

se observa las muestras del tipo B, empiezan en el valor 16, ya que las 15 primeras le corresponden a las muestras del tipo A.

Por el contrario las del tipo A, la red no las clasifica correctamente, ya que los valores no se aproximan a 1, como se puede observar en la ilustración siguiente.

1	0.2610	0.2417	0.1087	0.3140	0.1981	-0.0020	0.0848	0.1802	0.3347	0.3448
2	0.0384	0.0408	-0.0180	-0.0631	0.0191	0.0595	-0.0894	0.0349	-0.0291	0.0339
3	0.0919	0.1043	0.2834	0.1368	0.2074	-0.0204	0.0901	-0.0195	0.1841	0.1498
4	-0.0477	0.0286	0.0969	0.1847	0.2375	-0.2535	0.0301	0.1705	-0.0440	-0.0073
5	0.2447	0.2654	0.3790	0.4569	0.4937	0.1244	0.6016	0.1670	0.6509	0.3752
6	-0.1313	0.0037	0.0880	0.1930	-0.0120	0.0301	0.0341	-0.0299	0.1453	0.0522
7	0.1789	-0.0716	0.0180	0.1660	-0.1558	-0.1374	0.2431	-0.0645	-0.0054	0.2231
8	-0.0983	-0.1894	0.0413	-0.0373	-0.0829	0.1510	-0.1401	-0.0250	-0.2049	0.0555
9	0.0544	0.1194	-0.0967	0.0432	-0.0157	0.0503	0.1674	0.0104	0.1975	-0.0311
10	0.0880	-0.0181	0.0281	0.1160	0.1070	-0.2027	0.1110	-0.0278	0.1292	-0.0288
11	0.0539	0.1004	-0.0744	-0.2130	-0.0231	0.1326	0.0789	0.0290	-0.0248	-0.0157
12	0.2263	-0.0770	-0.0076	0.0281	0.0181	0.1615	0.1293	0.1157	0.1099	-0.2499
13	0.2005	0.0855	0.1992	-0.1036	0.4005	0.2666	0.0540	0.2281	-0.0219	0.0193
14	-0.0689	0.0368	0.1057	-0.0556	0.2058	0.0103	0.1087	-0.1361	0.0615	0.0613
15	-0.0689	-0.0314	0.1942	-0.2540	-0.2523	0.1420	0.0248	0.1707	-0.0609	-0.0084
16	-0.1962	0.2148	0.0701	-0.0957	-0.0992	0.1589	-0.0968	0.1359	0.1981	0.2867
17	0.0673	0.2340	0.1594	-0.1826	-0.1310	0.4389	-9.7572e-04	0.0552	-0.0374	0.1727
18	0.0683	0.1259	-0.0432	-0.0471	-0.0116	-0.0508	-0.0024	-0.0075	0.0028	-0.0654
19	0.2132	0.1386	-0.0829	0.3720	0.2546	0.1152	0.0686	0.2320	0.1989	0.0301

Ilustración 4. 4. Extracto de etiquetas de salida, simulación con 24 neuronas una única iteración, análisis de tipo roca A.

Con el fin de conocer el porcentaje de acierto de la red, se emplea la matriz de confusión, la cual nos indica la correspondencia entre los datos de salida y la etiqueta de entrada, mediante una matriz cuadrada.

A continuación se muestra un ejemplo de una matriz de confusión, donde existe una serie de tipos de rocas con un total de 15 muestras, mediante esta matriz se puede ver claramente donde la red confunde o clasifica erróneamente las muestras.

Matriz de confusión de tipos de rocas			
	Roca tipo A	Roca de tipo B	Roca de tipo C
Roca tipo A	5	0	0
Roca tipo B	0	3	2
Roca tipo C	0	2	3

Tabla 4. 2. Matriz de confusión de tipos de rocas, ejemplo explicativo de una matriz de confusión.

Como se puede observar en la tabla anterior, la red clasifica perfectamente las muestras del tipo A, mientras que las muestras B y C, la red no es capaz de clasificarlas correctamente confundiendo dichas muestras.

Para obtener el porcentaje de éxito de la red se debe de dividir el número total de muestra entre la sumatoria de la diagonal y multiplicarlo por cien.

$$Porcentaje\ de\ éxito\ de\ la\ red = \frac{\sum Diagonal}{N^o\ total\ de\ muestras} * 100 \quad (4.1)$$

Tabla 4. 3. Fórmula para obtener porcentaje de éxito a partir de la matriz de confusión

Se aplica la ecuación 4.1 sobre la matriz de confusión de ejemplo, obteniéndose que el porcentaje de éxito de dicha red sea del 73,33 %.

$$Porcentaje\ de\ éxito\ de\ la\ red = \frac{5 + 3 + 3}{15} * 100 = 73,33\ \% \quad (4.2)$$

Tabla 4. 4. Resultado de aplicar fórmula de porcentaje de éxito a partir de la matriz de confusión de ejemplo.

Así mismo se incluye una matriz de confusión donde se puede ver, los resultados de una de las simulaciones realizadas.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	4	0	0	0	7	0	0	0	0	0	0	0	0	0	0	1	2	0	1
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	2	0	1	0	2	1	0	0	0	0	0	1	2	1	2	0	1
4	0	0	0	9	0	0	1	0	0	0	0	0	1	1	3	0	0	0	0
5	0	0	0	0	13	1	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	14	0	0	0	0	1	0	0	0	0	0	0	0	0
7	1	0	0	1	1	0	4	0	0	0	0	2	1	1	0	4	0	0	0
8	0	0	0	0	0	0	0	8	4	3	0	0	0	0	0	0	0	0	0
9	0	0	0	3	0	0	0	2	9	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	0	0
11	1	0	1	0	2	0	1	0	0	2	7	0	0	0	0	0	1	0	0
12	0	0	0	0	2	0	0	0	0	0	0	11	0	0	0	2	0	0	0
13	1	0	0	1	0	0	1	0	0	0	0	3	7	0	0	1	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	2	0	1	0	0
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0
16	1	0	2	0	0	0	1	0	0	0	0	0	0	0	0	9	1	0	1
17	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	12	0	0
18	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	12	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14

Tabla 4. 5. Matriz de confusión para una red con 24 neuronas, y una única iteración.

En esta matriz de confusión se ve claramente que la red clasifica perfectamente, la muestra de tipo 2 y da unos resultados muy buenos para las muestras 15 y 19, como se ve

las muestras de estos tipos tienen características que las distinguen bastante bien del resto, ya que con una única iteración, la red las ha clasificado correctamente. Las muestras del tipo 2 o B, son las del hormigón visto, las cuales son superficies bastantes regulares, y que tiene una porosidad homogénea, estas características permiten este alto grado de acierto con respecto al resto. Siendo el porcentaje de acierto de esta red del 63,86 %.

En el siguiente esquema se puede ver el funcionamiento del algoritmo creado:

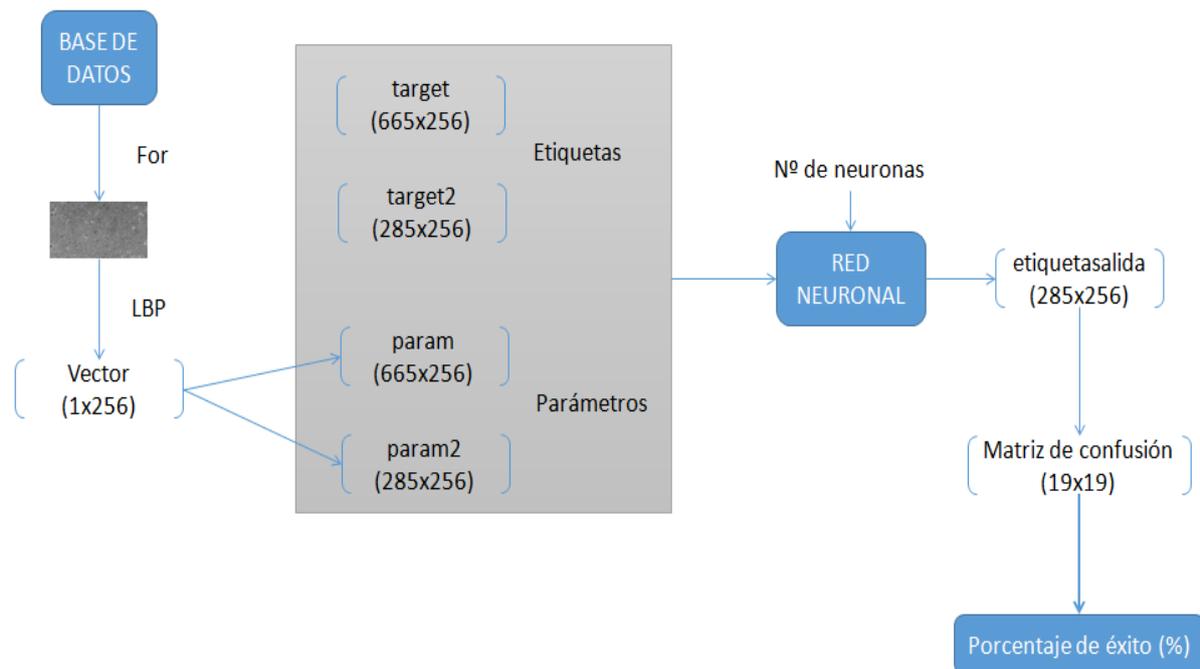


Ilustración 4. 5. Esquema del primer experimento.

#### 4.2.3. Segundo experimento: Extracción de LBP por nivel de DWT

En el siguiente experimento se aplicó sobre las imágenes de la base de datos, el *DWT* donde este método nos da un total de cuatro sub-imágenes, de las cuales la mayor información se encuentra en la sub-imagen obtenida al pasar el filtro de paso bajo-bajo, como se describió en el capítulo III. Sobre la imagen GG se le vuelve a pasar el *DWT* y se repite este procedimiento dos veces más, hasta obtener un total de 4 sub-imágenes de filtro paso bajo-bajo (GG, GG2, GG3 y GG4), siendo cada una de estas imágenes de distinto tamaño, ya que va disminuyendo, denominando a esta técnica arquitectura de aproximación directa hasta el cuarto nivel.

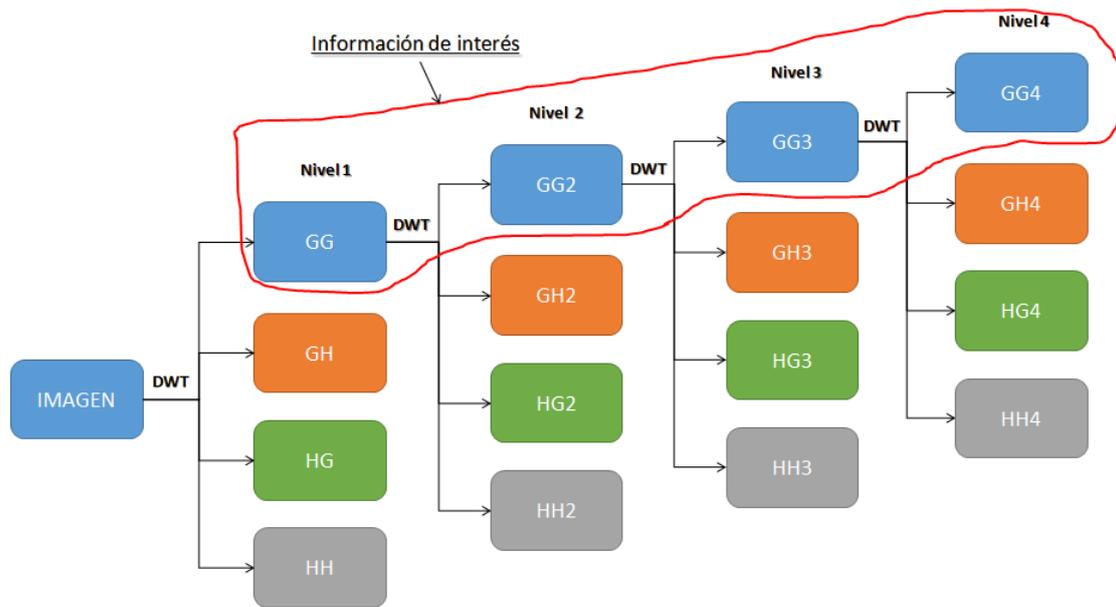


Ilustración 4. 6. Esquema del método DWT aplicando la arquitectura de aproximación directa hasta el 4º nivel.

Para poder trabajar con los datos anteriores, estos al ser de distinto tamaño, no se pueden trabajar directamente con ellos, por lo que a las imágenes obtenidas se le aplicó el LBP, con el fin de normalizar los datos y obtener las matrices de parámetros (*param* y *param2*) a emplear en la red neuronal.

256x1140 double

	1	2	3	4	5	6	7	8	9	10
236	20705	4825	1020	222	23697	5259	1080	214	26137	6001
237	5642	1912	603	169	5331	1988	532	165	1947	960
238	5070	1670	496	110	3124	1259	429	149	1810	872
239	3440	1433	498	193	3872	1691	529	153	1580	917
240	29713	9755	2783	790	39630	11993	3051	805	16858	7672
241	49766	8424	1346	258	76928	10976	1751	291	77255	12557
242	4762	1536	412	87	6650	1895	403	117	6400	1972
243	2554	894	258	49	1709	701	212	67	1880	710
244	3395	1318	423	102	3909	1417	472	141	2380	1135
245	29268	5262	852	152	32770	6093	975	155	40599	7202
246	4031	1256	346	81	3918	1208	348	85	6881	2113
247	21439	4940	966	200	20986	5048	1036	191	12714	4011
248	29914	9965	2749	768	40590	12098	3021	830	16917	7524
249	36684	7153	1567	342	39940	7314	1485	309	40512	7537
250	19649	4632	1043	196	17392	4364	974	219	29365	6381
251	1839	799	329	80	1146	494	255	74	1858	709
252	12059	4075	1197	302	12148	3779	1183	307	19621	5741
253	20902	4725	955	201	13904	3916	999	215	15127	3914
254	16559	4030	988	248	7711	2733	862	215	20313	4990

Ilustración 4. 7. Extracto de la matriz de parámetros, param2 (256x1140).

Tras esto, como se hizo en el experimento anterior, se crean las matrices de etiquetas (*target* y *target2*).

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0

Ilustración 4. 8. Matriz de etiqueta, target2 (19x1140).

Una vez la red se entrena y testea, se obtiene una matriz de salida (etiquetasalida), de la cual se calcula la matriz de confusión, y se consigue el porcentaje de éxito de la red.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	46	0	1	0	3	0	1	1	0	0	0	0	3	0	0	0	1	0	4
2	0	56	1	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
3	3	0	9	2	0	2	1	0	3	1	2	0	2	3	18	12	2	0	0
4	0	3	0	29	0	0	6	0	3	0	2	1	2	2	5	4	2	1	0
5	2	0	0	0	46	3	0	0	0	0	7	0	1	0	0	1	0	0	0
6	2	1	0	0	3	43	0	0	1	0	3	2	0	2	0	1	0	0	2
7	6	0	4	5	1	0	4	0	0	2	1	5	22	3	3	2	1	0	1
8	0	0	0	0	0	0	0	49	5	6	0	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	0	1	52	0	0	0	1	0	1	3	1	0	0
10	0	0	0	0	1	0	0	4	0	55	0	0	0	0	0	0	0	0	0
11	2	0	0	0	9	3	1	0	0	4	38	0	1	0	0	2	0	0	0
12	1	3	1	2	0	0	3	0	0	1	0	42	5	0	0	1	0	1	0
13	0	2	1	2	0	0	2	0	0	0	0	4	45	0	2	0	2	0	0
14	2	0	1	1	1	1	1	0	0	0	0	1	4	42	4	0	2	0	0
15	0	0	0	4	0	0	1	0	6	0	3	0	1	0	41	2	1	1	0
16	0	1	2	2	0	1	1	1	1	0	1	1	5	1	7	32	4	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	59	0	0
18	0	0	0	0	1	0	0	0	2	0	0	2	0	0	0	0	1	51	3
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Tabla 4. 6. Matriz de confusión de la etiquetasalida para una red con 28 neuronas, y 12 iteraciones, en el segundo experimento.

En esta matriz de confusión se debe de tener en cuenta que el número de muestras aumenta, donde antes se tenía solo 15 muestras, ahora se tiene un total de 60, ya que se obtiene 4 imágenes por muestra ( $15 \cdot 4 = 60$ ).

Se ve claramente que la red clasifica perfectamente, la muestra de tipo 19 y da unos resultados muy buenos para las muestras 2 y 17. Siendo el porcentaje de acierto de esta red del 70,09 %.

En el siguiente esquema se puede ver cómo funciona el experimento descrito anteriormente.

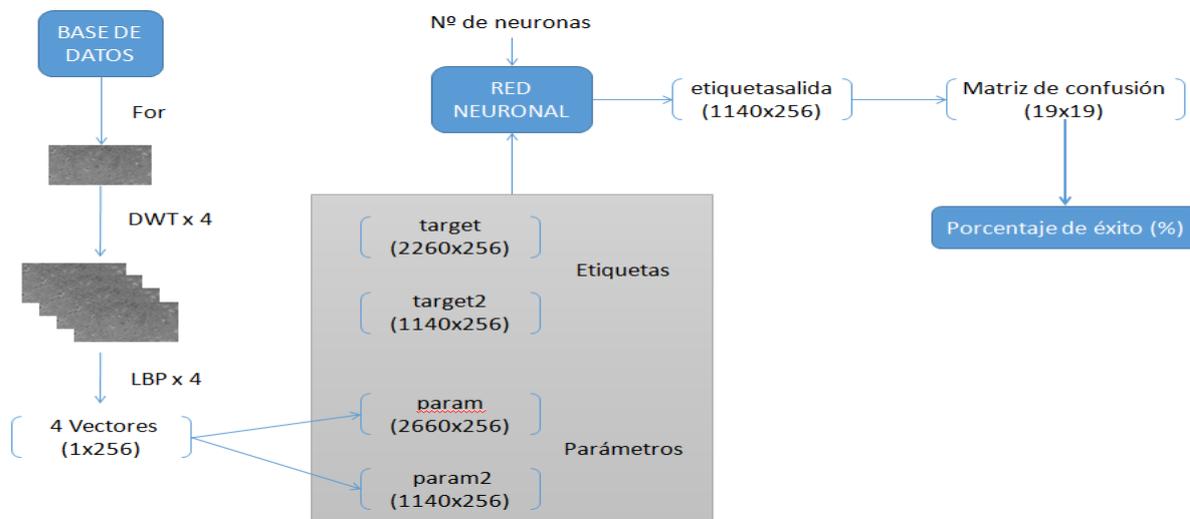


Ilustración 4. 9. Esquema del segundo experimento.

NOTA:

Como se puede ver en el esquema anterior, la red neuronal trabaja con matrices cuatro veces mayores a las del primer experimento, **esto supone un incremento del coste computacional, y por ende mayor tiempo en el procesado.**

#### 4.2.4. Tercer experimento: DWT + LBP conjuntamente

El siguiente experimento es muy parecido al anterior, pero en esta ocasión, una vez se extrae el *LBP* de cada sub-imagen (GG, GG2, GG3 y GG4), estos se concatenan en un único vector, obteniendo un vector *LBP* formado por 1 fila con 1024 ( $256 \cdot 4$ ) columnas.

Variables - LBP													
LBP													
1x1024 double													
	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024
1	173	183	79	202	907	334	176	65	322	166	176	297	2603

Ilustración 4. 10. Extracto de vector LBP, concatenando los resultados LBP de GG, GG2, GG3 y GG4.

Los pasos siguientes son exactamente iguales a los de los experimentos anteriores, se crean las matrices de etiqueta (*target* y *target2*), se obtienen los parámetros (*param* y

*param2*), se calcula la red neuronal, se obtiene unos resultados de salida (etiquetasalida), y se calcula la matriz de confusión para obtener el porcentaje de éxito de la red.

Variables - param										
param										
665x1024 double										
	1	2	3	4	5	6	7	8	9	10
1	45422	8724	2531	8853	4541	895	5469	20331	28216	12766
2	89921	10192	6814	10071	12803	1569	11527	20182	41557	11224
3	60495	5648	14395	10433	11748	1201	15997	19365	14266	8879
4	63261	9387	8723	10905	8828	1283	10993	18418	21429	11158
5	52220	5726	8370	7667	14612	1551	15795	18922	14157	8352
6	87187	21289	22019	13921	5706	1491	8426	8483	14193	13221
7	62095	8909	8936	9629	7106	1280	8780	10794	20391	9842
8	108526	15333	21701	23204	12852	2374	21704	28072	31204	24964
9	93809	12405	14477	19496	12528	1748	20235	34437	33792	22369
10	68552	10956	9181	14113	7782	1191	12167	20941	25882	15560
11	118437	19186	20051	20707	12561	2365	17418	24503	34706	21539
12	104097	16582	16401	19992	10134	1501	16300	26259	34714	22374
13	106609	19956	25051	23166	8961	1849	16507	23651	24513	22785
14	119884	25128	24683	21330	9220	2274	14117	20984	27533	21539
15	110936	13818	21503	19453	15523	2175	21117	26999	30810	19922
16	107149	14328	20391	20471	14022	1954	19582	27386	29237	21401
17	94480	17429	20678	24260	10440	2034	17630	30713	26685	24125
18	91695	15159	18267	22140	10879	1567	18460	31029	26879	23076
19	115129	13568	16877	16943	16608	2329	19291	29954	38906	19132

Ilustración 4. 11. Matriz de parámetros, *param* (665x1024), para el tercer experimento.

Variables - target										
target										
19x665 double										
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0

Ilustración 4. 12. Matriz de etiqueta, *target* (19x665), para el tercer experimento.

En este experimento se realizó una variación respecto al anterior, previamente antes de pasar a realizar las simulaciones se probó con varias familias de funciones de *Wavelet*, con el fin de ver con cual se conseguía un mejor resultado, siendo estas familias las siguientes:

- Haar, conocida como el primer *Wavelet*, esta está formada por un conjunto de funciones, siendo el más simple de todos, su principal desventaja que no es derivable, empleándose normalmente en el análisis de transiciones repetidas.
- Bior4.4, esta función permite la reconstrucción de la señal, a diferencia del resto de familiar, se caracteriza por su fase lineal.
- Sym4, esta familia se trata de una modificación de la *Daubechies Wavelet*, esta permite análisis de multi-resolución ortogonal.

A continuación se muestra los filtros empleados por los algoritmos de Matlab, obtenidos de la página oficial, la cual se puede consultar en la siguiente dirección url: <https://es.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html#f3-1009152>.

- Haar:

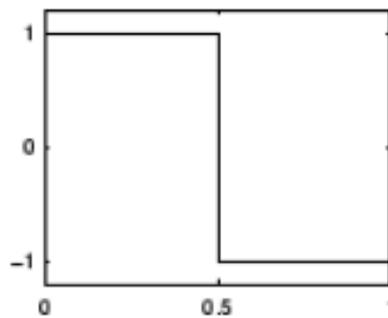


Ilustración 4. 13. Filtro de Haar.

- Bior4.4:

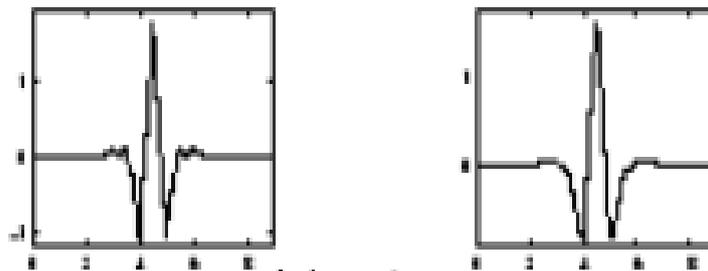


Ilustración 4. 14. Filtro bio-ortogonal 4.4.

- Sym4:

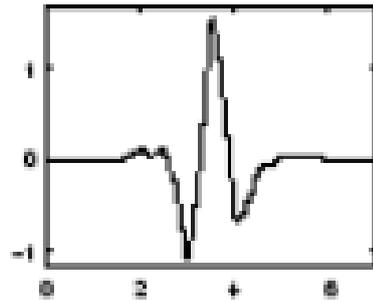


Ilustración 4. 15. Filtro Symlet 4.

El comando para obtener el *DWT* en Matlab es el siguiente:

```
[cA,cH,cV,cD] = dwt2(filename, 'wavelet'); % en el apartado wavelet, se
sustituye por la familia a aplicar, 'haar', 'bior4.4' o 'Sym4'.
```

En el siguiente esquema se puede ver cómo funciona el experimento descrito anteriormente.

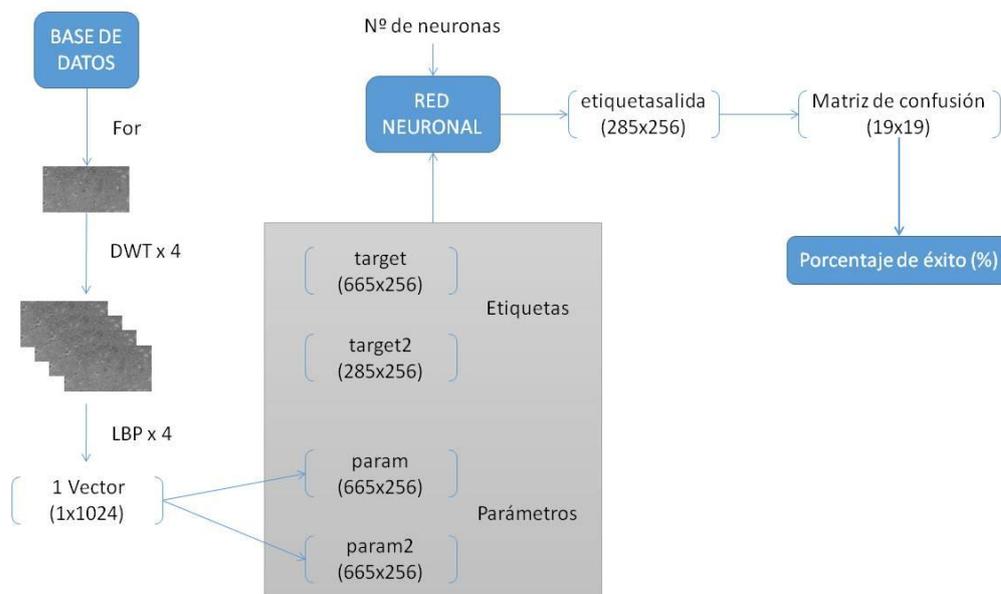


Ilustración 4. 16. Esquema del tercer experimento.

### 4.3. RESULTADOS

#### 4.3.1. Experimento 1: LBP

En primer lugar se realizó la clasificación de la red con distintos números de neuronas, con el fin de ver el nº óptimo de neuronas a emplear para entrenar la red, donde se vio que la red neuronal funcionaba mejor con 24 neuronas como se puede observar en el gráfico siguiente:

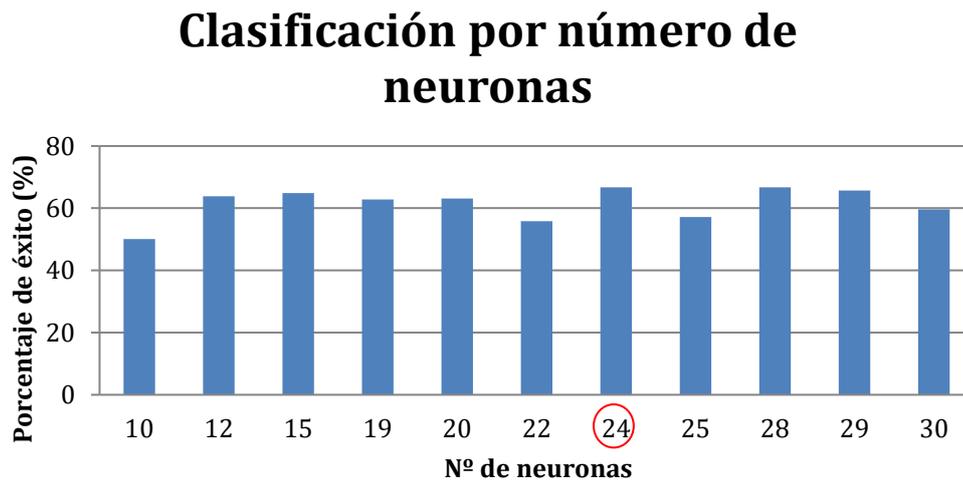


Ilustración 4. 17. Gráfica de clasificación por nº de neuronas para el primer experimento.

Sobre los resultados anteriores se calculó el promedio de las etiqueta de salida, se aplica la ecuación 4.1 y se obtiene el porcentaje de éxito, obteniendo 69,12 % de éxito.

Cuya matriz de confusión nos da los siguientes resultados.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	6	0	0	0	2	0	0	0	0	0	0	0	4	0	0	0	3	0	0
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	5	0	0	0	1	1	0	0	0	0	0	1	2	1	2	0	2
4	0	0	0	10	0	0	1	0	0	0	0	0	0	1	3	0	0	0	0
5	0	0	0	0	11	3	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	13	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	1	1	0	2	0	0	0	0	3	3	2	0	2	1	0	0
8	0	0	0	0	0	0	0	11	3	1	0	0	0	0	0	0	0	0	0
9	4	0	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
11	2	0	2	0	2	0	0	0	0	0	7	0	1	0	0	0	1	0	0
12	0	0	1	0	2	0	0	0	0	0	0	10	0	0	0	2	0	0	0
13	0	0	0	1	0	0	0	0	0	0	0	3	10	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	3	0	0	0	0
15	0	0	0	2	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
16	2	0	0	1	0	0	0	0	0	0	0	0	1	1	2	7	1	0	0
17	1	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	11	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14

Tabla 4. 7. Matriz de confusión del promedio de etiquetas de salidas, primer experimento.

Como se puede observar en la matriz anterior, la red clasifica perfectamente de los 19 tipos de rocas 3 de ellas, siendo estos los tipos 2, 10 y 18 respectivamente. Confundiendo el resto en mayor o menor medida, como por ejemplo la red confunde una de las muestras del tipo 19 con la del tipo 17.



Ilustración 4. 18. Muestra de tipo 19 (izquierda) y 17 (derecha).

Se observa que las imágenes anteriores, la red no debería de confundir dichas muestras, esto es debido a que la red entrena la capa oculta con pesos aleatorios, los cuales pueden ser más acertados o menos, por ello es necesario trabajar sobre la red de manera iterativa, empleando los mismo datos (datos de entrada y número de neuronas) con el fin de que vaya ajustando los pesos a medida que trabaja con ellos, mediante este

procedimiento se simula la mecánica del aprendizaje de las redes neuronales en los seres humanos.

Por ello el siguiente pasó que se realiza, es trabajar con redes neuronales, donde se realizan simulaciones con diverso número de neuronas y se aplican diversas iteraciones, con el fin de ver como mejora los resultados, los cuales se explican a continuación.

- 1) **Entrenamiento con 10 neuronas:** el mejor resultado de la red es con 22 iteraciones, obteniendo un porcentaje de éxito del 66,67 %, los datos de éxito de las simulaciones siguientes son inferiores.

### Porcentaje de Éxito (%) con 10 neuronas

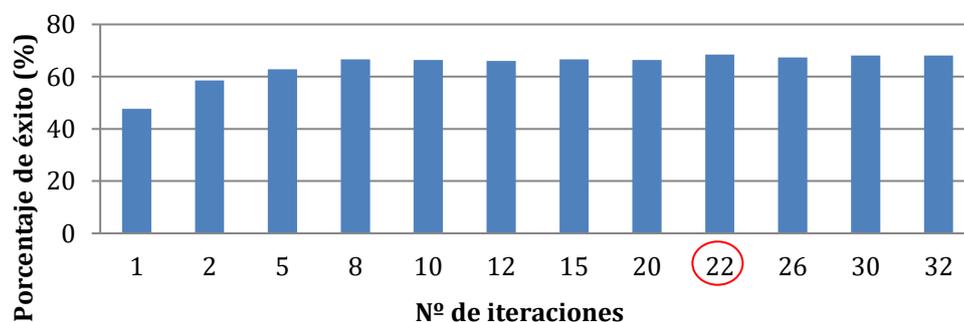


Ilustración 4. 19. Gráfica del porcentaje de éxito de una red con 10 neuronas ocultas.

- 2) **Entrenamiento con 24 neuronas:** el mejor resultado de la red es con 10 iteraciones, obteniendo un porcentaje de éxito del 70,53 %.

### Clasificación con 24 neuronas

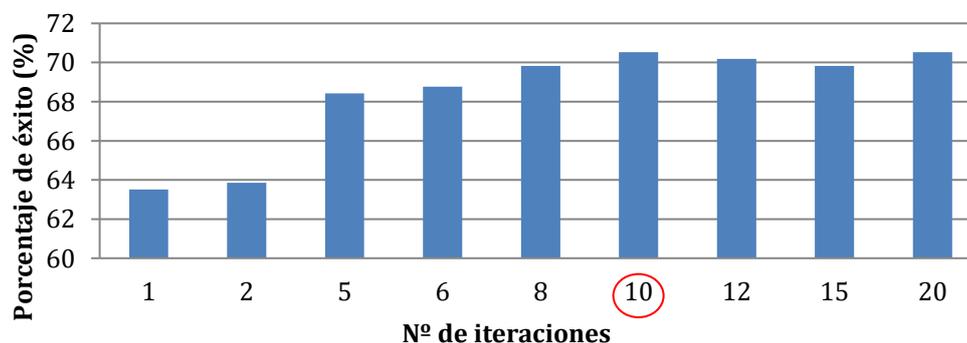


Ilustración 4. 20. Gráfica del porcentaje de éxito de una red con 24 neuronas ocultas

- 3) **Entrenamiento con 28 neuronas:** el mejor resultado de la red es con 12 iteraciones, obteniendo un porcentaje de éxito del 70,88 %.

## Clasificación con 28 neuronas

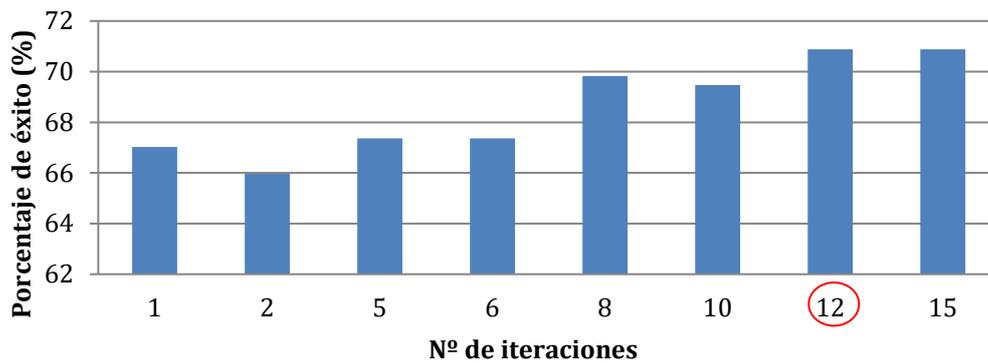


Ilustración 4. 21. Gráfica del porcentaje de éxito de una red con 28 neuronas ocultas

- 4) **Entrenamiento con 30 neuronas:** el mejor resultado de la red es con 15 iteraciones, obteniendo un porcentaje de éxito del 70,88 %.

## Clasificación con 30 neuronas

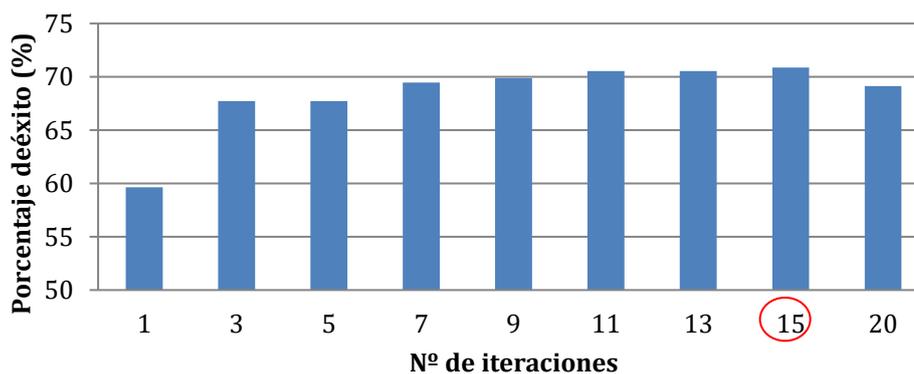


Ilustración 4. 22. Gráfica del porcentaje de éxito de una red con 30 neuronas ocultas

De los resultados anteriores se desprende, que la red se comporta mejor, con 24 neuronas, donde con 10 iteraciones se obtiene su mejor resultado, decreciendo a medida que aumenta el número de iteraciones y nº de neuronas, por ende el coste computacional, es decir, más horas de cálculo con peores resultados.

Siendo la matriz de confusión obtenida para esta simulación la que se muestra a continuación.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	4	0	0	0	3	0	0	0	0	0	0	0	5	0	0	0	3	0	0
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	6	0	0	0	1	1	0	0	0	0	0	1	2	1	2	0	1
4	0	0	0	10	0	0	1	0	0	0	0	0	1	0	3	0	0	0	0
5	0	0	0	0	12	2	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	13	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	1	2	0	4	0	0	0	0	3	3	1	0	0	1	0	0
8	0	0	0	0	0	0	0	10	4	1	0	0	0	0	0	0	0	0	0
9	4	0	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
11	1	0	2	0	2	1	0	0	0	0	6	0	1	0	0	1	1	0	0
12	0	0	1	0	2	0	0	0	0	0	0	10	0	0	0	2	0	0	0
13	0	0	0	1	0	0	0	0	0	0	0	3	11	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	3	0	0	0	0
15	0	0	0	2	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
16	1	0	3	0	0	0	0	0	0	0	0	0	0	1	1	8	1	0	0
17	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	13	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14

Tabla 4. 8. Matriz de confusión de etiquetas de salida, 24 neuronas con 10 iteraciones primer experimento.

Como se puede observar en la matriz anterior, la red clasifica perfectamente los tipos 2, 10 y 18, como en la matriz de confusión de las etiquetas de salida de la tabla 4.7 descrita anteriormente.

Tras esto, se analiza los datos promedios de la red, con el objeto de ver si mejora o empeora el porcentaje de éxito, en la siguiente grafica se puede ver los porcentajes por número de iteraciones:

Nº Iteraciones	Porcentaje de Éxito (%)
1	63,51
2	63,86
5	68,42
6	68,77
8	69,82
10	70,53
12	70,18
15	69,82
20	70,53
22	70,18
23	70,53
25	70,18
26	69,47
30	70,18

Tabla 4. 9. Porcentaje de éxito según nº de iteraciones, para 24 neuronas.

Se realiza un total de 14 simulaciones, cuyo porcentaje de éxito promedio es del 69,12 % y cuya matriz de confusión obtenida se muestra a continuación:

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	4	0	0	0	3	0	0	0	0	0	0	0	5	0	0	0	3	0	0
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	7	0	0	0	1	1	0	0	0	0	0	1	1	2	2	0	0
4	0	0	0	10	0	0	1	0	0	0	0	0	1	0	3	0	0	0	0
5	0	0	0	0	12	2	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	13	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	1	1	0	2	0	0	0	0	3	3	2	0	2	1	0	0
8	0	0	0	0	0	0	0	10	4	1	0	0	0	0	0	0	0	0	0
9	4	0	0	0	0	0	0	1	10	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	0	0
11	1	0	2	0	3	1	0	0	0	0	6	0	1	0	0	0	1	0	0
12	0	0	0	0	2	0	0	0	0	0	0	10	0	0	0	3	0	0	0
13	0	0	1	1	0	0	0	0	0	0	0	3	10	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	3	0	0	0	0
15	0	0	0	2	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
16	1	0	4	0	0	0	0	0	0	0	0	0	0	1	1	7	1	0	0
17	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	13	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14

Tabla 4. 10. Matriz de confusión promedio de etiquetas de salida 24 neuronas, primer experimento.

Como se ve los datos promedios empeoran los resultados, con respecto a la red con 10 iteraciones, con lo cual se puede decir que el mejor resultado obtenido en este experimento, es la red neuronal donde se emplean 24 neuronas y 10 iteraciones, que empeoren los resultados se debe a que a medida que se aumenta el número de iteraciones, se obtienen porcentajes de éxitos inferiores, lo cual al promediar tiende a bajar y no a subir.

#### 4.3.2. Experimento 2: Extracción de LBP por nivel de DWT

Este experimento aunque pueda parecer el más corto, se ocupó mayor tiempo que el resto de experimentos, a la hora de realizar las simulaciones, debido a que el tiempo medio por simulación solo con 10 neuronas y una única iteración, comprendía entorno a unas 2-3 días.

En la tabla siguiente se puede ver la comparativa entre ambos experimentos:

Comparativa entre el experimento 1 y 2				
Nº de Neuronas	Nº de Iteraciones	Porcentaje de éxito (%) con <i>DWT</i>	Porcentaje de éxito (%) sin <i>DWT</i>	Diferencia (%)
10	1	46,49	47,72	-1,23
24	1	63,6	63,51	0,09
24	10	68,07	70,53	-2,46
28	12	70,09	70,88	-0,79

Tabla 4. 11. Comparativa de resultados de éxito entre los dos primeros experimentos.

Como se puede ver en la tabla anterior, los resultados que se obtienen disminuye el porcentaje de éxito, esto es debido a que:

- El número de neuronas empleado no es suficiente para los datos de entrada, ya que cuadriplica con respecto al experimento anterior.
- Los tiempos de cálculo de la red son mucho mayores a los del experimento anterior.

Por todo ello, este experimento para la clasificación se descarta y se propone un tercer experimento, con el cual comparar los datos del experimento anterior, con el fin de ver cuál de ellos es mejor para realizar la clasificación de rocas por imágenes.

#### 4.3.3. Experimento 3: DWT + LBP conjuntamente

Este experimento es muy similar al primero, pero se tendrá un conjunto de datos de entrada mayor, permitiendo mejorar la clasificación de la red.

Previamente se realiza un estudio con el fin de ver qué familia *Wavelet* se adaptaba mejor a los datos de entrada, para ello se realizó varias simulaciones, donde se aplican los filtros correspondientes por familias, posteriormente se introducen en la red neuronal y se obtiene los resultados éxito de la red, siendo estos los que aparecen en la tabla siguiente:

Evaluación de familias wavelet			
Nº de Neuronas	Numero Iteraciones	FAMILIA	Porcentaje de Éxito (%)
10	1	HAAR	36,84
10	1	SYM4	<b>49,12</b>
10	1	BIOR4.4	43,51
10	5	HAAR	56,49
10	5	SYM4	<b>72,63</b>
10	5	BIOR4.4	65,26
15	1	HAAR	48,07
15	1	SYM4	<b>61,4</b>
15	1	BIOR4.4	47,02

Tabla 4. 12. Evaluación de las familias Wavelet.

Los mejores resultados se consiguen con la familia de *Wavelet Sym4*, siendo esta la empleada para el presente experimento.

Al igual que en el experimento primero, se crea diversas simulaciones por número de neuronas y variando el número de iteraciones, en la siguiente tabla se muestra los resultados de dichas simulaciones.

Nº de Neuronas	Nº de Iteraciones	Porcentaje de éxito (%)
10	10	77,19
10	15	76,14
15	10	78,95
17	1	64,21
18	1	68,07
18	10	80,7
18	15	81,4
18	20	80,35
18	25	81,75
18	26	81,4
18	30	81,05

Nº de Neuronas	Nº de Iteraciones	Porcentaje de éxito (%)
20	1	63,86
20	10	80
20	20	80,35
20	30	81,75
21	1	48,07
21	5	64,21
21	10	80,35
21	15	81,75
21	20	79,3

Tabla 4. 13. Resultados de porcentaje de éxito en el experimento tercero.

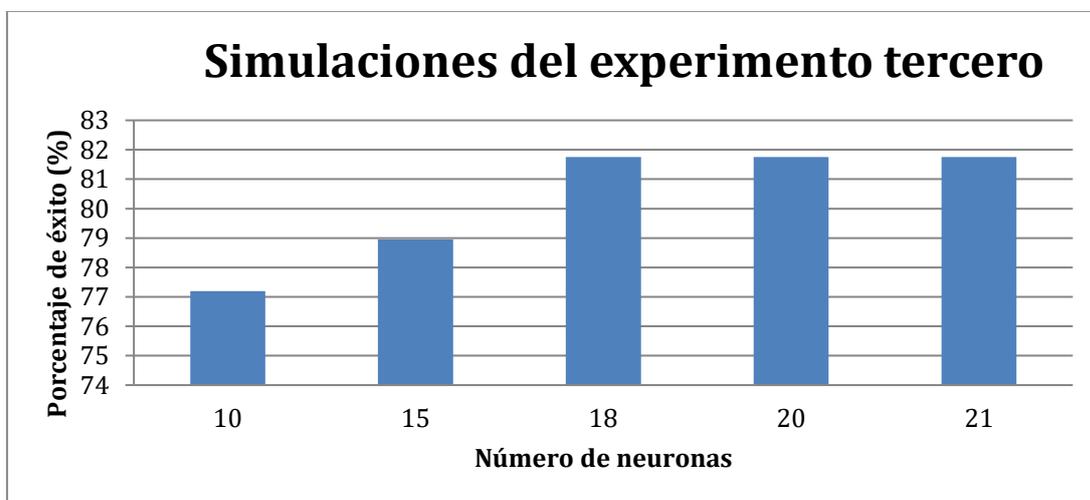
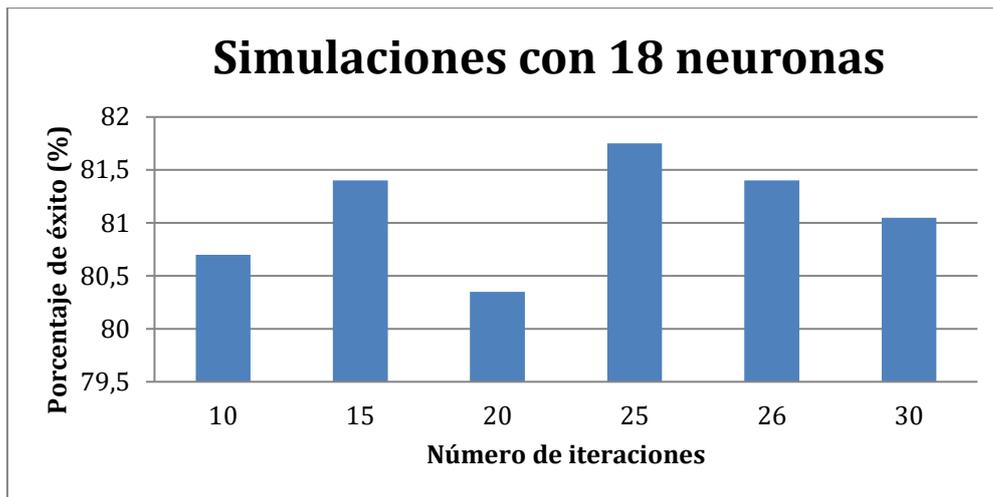


Ilustración 4. 23. Mejor resultado por nº de iteraciones, tercer experimento.

El grafico anterior nos muestra los mejores resultados del tercer experimento, donde se observa que la red independientemente del número de neuronas, a partir de 18 neuronas, el porcentaje de éxito de la red, no superar el 81,75 %, por lo tanto la red se mantiene estable, sin mejorar el porcentaje de éxito.

Finalmente el mejor resultado de las simulaciones se obtiene para una red con 18 neuronas y 25 iteraciones, en el grafico siguiente, se comprueba que la red a medida que aumenta el número de iteraciones, disminuye el porcentaje de éxito, por lo tanto el valor óptimo de la red es con 24 neuronas y 25 iteraciones.



Cuya matriz de confusión que se obtiene es la siguiente.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	8	1	0	0	2	0	0	1	1	0	0	0	1	1	0	0	0
4	0	0	0	12	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0
5	0	0	0	0	13	2	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	14	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	3	0	0	0	0	2	7	0	2	1	0	0	0
8	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
11	0	0	0	0	2	1	0	0	0	3	8	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	1	0
13	0	0	0	3	0	0	0	0	0	0	0	2	10	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
16	1	0	2	2	0	0	0	0	0	0	0	0	1	0	1	8	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15

Tabla 4. 14. Matriz de confusión de la etiqueta de salida 18 neuronas y 25 iteraciones, tercer experimento.

Como se muestra el porcentaje de éxito con respecto al primer experimento, es bastante mayor, clasificando la red perfectamente un total de 7 sobre 19 tipos de rocas. Analizando la red se aprecia que la mayor confusión a la hora de clasificar las muestras, es con las muestras del tipo nº 7.



Ilustración 4. 24. Muestra del tipo de roca nº 7.

Como se puede observar en la imagen anterior, este tipo de roca tiene una rugosidad uniforme, generando que la red clasifique erróneamente este tipo de rocas, ya que se parece mucho a otros tipos por su uniformidad. Si se ve la matriz de confusión esta red confunde mucho con la roca nº 13, confundiendo 7 de las muestras con este tipo.

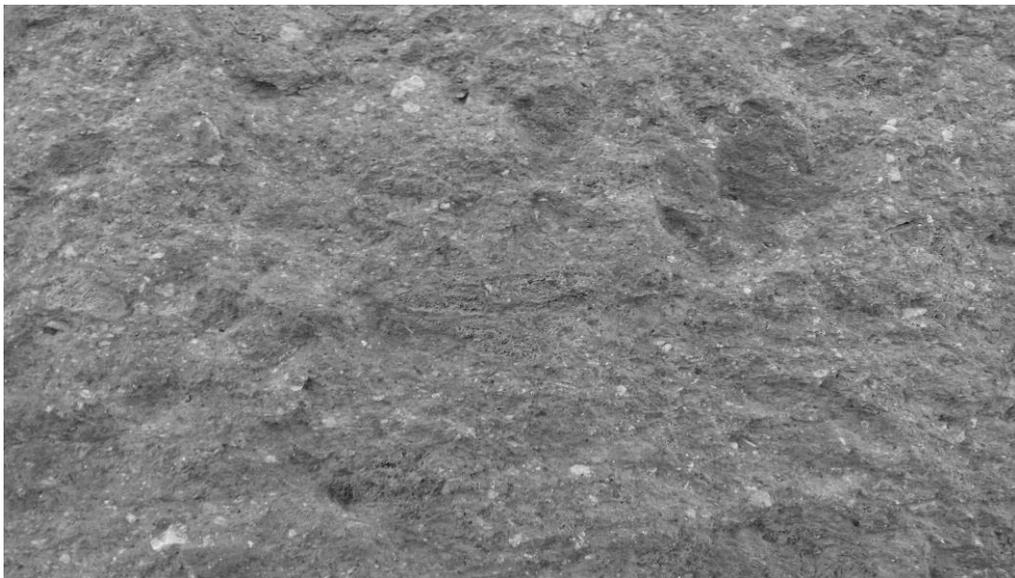


Ilustración 4. 25. Muestra del tipo de roca nº 13.

Se ve que la imagen anterior es muy similar al nº 7, por lo tanto, es lógico que la red confunda estos tipos. Aunque por otro lado la red clasifica casi perfectamente el tipo nº

13, confundiendo dos de sus muestras con el tipo nº 12, siendo una de las posibles muestras que se confunde, la que se muestra a continuación.

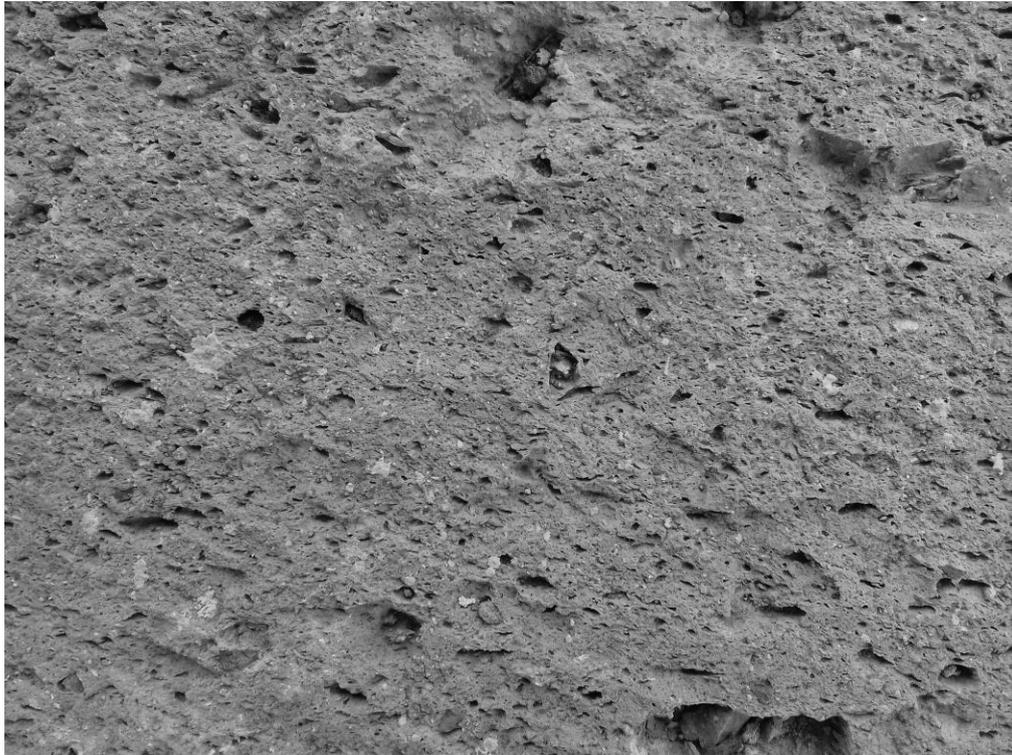


Ilustración 4. 26. Muestra del tipo de roca nº 13.

Este tipo de roca tiene mayor rugosidad aunque algunas de las tomas realizadas, son en zonas erosionadas, las cuales se pueden confundir con otros tipos de rocas.

Con el fin de disminuir aún más el porcentaje de fallo de la red, se combina todos los datos de salida de las simulaciones realizadas con 18 neuronas, y se calcula la matriz de confusión para los datos promedios, siendo esta la que se muestra a continuación.

Clase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	5	0	0	0	1	0	0	1	2	0	0	0	1	1	0	0	1
4	0	0	0	13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	13	2	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	3	0	0	0	0	1	7	0	3	1	0	0	0
8	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
11	0	0	0	0	3	0	0	0	0	4	8	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	1	0
13	0	0	0	3	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
16	1	0	3	1	0	0	0	0	0	0	0	0	1	0	1	8	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15

Obtenido un porcentaje de éxito del 82,46 %, mejorando el resultado en 0,71 %, consiguiendo que se clasifique perfectamente un tipo más de roca, siendo este tipo el nº 6, que en la matriz de confusión anterior era casi perfecta.

Tras todo debido a la falta de tiempo, no se desarrolló la clasificación por máquina de soporte vectorial, en adelante MSV, pero si se comprobó conforme a los datos obtenidos en una MSV sobre un kernel lineal, consiguiendo porcentajes de éxito del 21%, muy inferiores a los obtenidos con la red neuronal, este algoritmo fue proporcionado por el tutor.

Esto porcentaje de éxito tan bajo se debe a que la información contiene mucho ruido, y las MSV trabajan mal con datos ruidosos, en cambio las RNA, mediante los pesos descartan o minimizan los datos con más ruido, permitiendo conseguir clasificar los tipo de roca mucho mejor.

#### 4.4. COMPARATIVA CON EL ESTADO DEL ARTE

En la siguiente tabla se muestra un resumen del capítulo primero, del presente trabajo, donde se muestra el método empleado, número de muestras y porcentajes de éxitos:

Método de clasificación	Número de muestras por tipo de roca	Precisiones (%)
Textural (Vecino más próximo - función de discriminación lineal)	243	89
Algoritmo de clasificación supervisada y reglas de decisión de mínima distancia (Matriz de co-ocurrencia, textura espectral y espaciado textural)	58	89
<b>ANN - Microscopio de luz polarizada</b>	44 - [Entrenamiento de la red, 36 y 74 muestras por mineral]	93
<b>ANN</b>	> 1000, 2 entrenamientos de la red, una con 256 y otra con 215 de imágenes de distintas zonas	93,3-93,5
<b>ANN</b>	3 imágenes de minerales enriquecidos (Manganeso, Hierro, Alúmina, Agregados), de los tres minerales para identificar, empleando las estadísticas de las misma en el entrenamiento de la red.	88,71
Representación de Imágenes por Cúmulos Coordinados (RICC) - Clasificador por distancia mínima	36 imágenes, 8 clases, 3 imágenes por clase	N/S
390 imágenes de distinta iluminación y ángulo de captura		81-98
<b>ANN</b>	300 secciones finas	92,22
<b>Support Vector Machine (SVM)</b>	Base de datos con 530 imágenes	RMSE (29,8%)
<b>TSecSoft (Pre procesado - Segmentación - Etiquetado)</b>	Base de datos de la aplicación	N/S
<b>Support Vector Machine (SVM)</b>	1200 imágenes, 10 imágenes por muestra, con distintas orientaciones y lugar.	96,2

Método de clasificación	Número de muestras por tipo de roca	Precisiones (%)
<b>Algoritmo de agrupación por color (Segmentación mineral inteligente)</b>	155 muestras de 28 minerales diferentes.	92,15-85,24
<b>Majority Voting Schema (MVS)</b>	45 muestras de sección fina, 30 de plutónicas y 15 extrusivas e hiperbisal.	98
<b>ANN- segmentación e identificación</b>	135 imágenes de sección fina, 23 muestras de minerales de rocas ígneas,	93,81

Tabla 4. 15. Resumen de porcentaje de éxito de estado del arte.

Como se ve los porcentajes de éxito son bastantes elevados, esto es debido los siguientes motivos:

- Métodos de extracción de características más concretas, ya que se tiene conocimientos más profundos de la materia, y van directo a obtener los datos más característicos de esos tipos de rocas.
- Las imágenes se obtienen en condiciones óptimas (en laboratorio, con cámaras de alta resolución e iluminación uniforme).
- Previo pre procesado de las imágenes.

En nuestro caso:

- Se emplea cámara de 13 megapíxeles, aunque es una resolución buena, la calidad de la toma no es la misma que una cámara profesional, empleando siempre el mismo móvil para la captura de imágenes.
- Poca estabilidad, movimientos involuntarios de las manos en el momento de la toma, generando ruido, pudiendo evitarse si se emplea trípode.
- Imágenes en distintas horas del día, generando variación lumínica.
- De la misma roca no se realiza la toma en un único ángulo, sino que se toman muestras de distintas zonas, donde puede existir variaciones y ser más complicada la clasificación para la red.
- No se realiza pre procesado de las imágenes capturadas.

Como se observa en comparación con los resultados obtenidos en el estado del arte, nuestros resultados son peores, aunque al realizarse en las condiciones descritas, estos resultados son buenos, ya que si las mismas muestras se toman teniendo en cuenta las consideraciones anteriores, los porcentajes de éxitos de la red aumentarían considerablemente.

# CAPÍTULO V: CONCLUSIONES Y LINEAS FUTURAS



## 5.1. CONCLUSIONES

Con la finalización del presente TFM se ha cumplido los objetivos propuestos en un principio y consiguiendo porcentajes de éxito del 82,46 % en la identificación de rocas capturadas mediante dispositivos móviles.

La realización del presente trabajo ha permitido comprender mejor el uso de la inteligencia artificial para resolver problemas complejos, donde la aplicación de esta ciencia permite obtener unos resultados que sirven para contrastar hipótesis o dar una segunda opinión a personas cualificadas en una determinada materia. Así como permitir a la población en general, disponer o usar datos de su interés. Esta ciencia se aplica en otras áreas como puede ser la medicina, en el reconocimiento de melanomas, en la seguridad para el reconocimiento facial, etc.

Con lo dicho anteriormente, se consigue comprender mejor el uso y aplicación de dicha ciencia, más concretamente las redes neuronales, las cuales permiten obtener unos resultados usando unos datos de entrenamiento y validando los mismos mediante testeo, donde indistintamente del elemento u objeto a clasificar el procedimiento a seguir siempre será el mismo, se dispone de unos datos de partida, se extraen sus características y se pasan por la red neuronal.

En primer lugar indicar que los resultados obtenidos no son los esperados, pero estos tampoco han sido malos, ya que si se tiene en cuenta las precauciones a la hora de obtener los datos de partida, como se expuso en el capítulo anterior apartado cuarto, estos mejorarían. Se plantea en un inicio que con la resolución actual de las cámaras integradas en los dispositivos móviles, estos eran adecuados para la toma de datos, y aunque se procura realizar la toma con la mayor estabilidad posible, al no disponer de elementos de fijación, se produce mayor cantidad de ruido por movimientos involuntarios. Se debe tener en cuenta que en el estado del arte siempre las muestras se toman en condiciones de laboratorio, en este caso las condiciones era al aire libre, en distintas franjas horarias, y sin pre-procesado de las imágenes, por lo que los resultados obtenidos son un poco peores.

En segundo lugar el equipo de trabajo para las simulaciones, no era el más idóneo, ya que la capacidad del mismo no permite la realización de mucha de las simulaciones planteadas, empleando para ello otros equipos más potentes para poder llevar a cabo el

trabajo, esto produce demoras en el tiempo para realizar el mismo, ya que se depende de la disponibilidad de estos.

En tercer y último lugar los experimentos realizados, fueron un total de tres, aplicando únicamente las redes neuronales, ya que el segundo experimento al cuadruplicar el número de datos, consumió mucho tiempo material en las simulaciones, esto no permite aplicar otro método de clasificación como se propuso en un principio, que era la máquina de soporte vectorial.

## 5.2. LINEAS FUTURAS

De las conclusiones anteriormente expuestas se puede destacar las siguientes líneas de trabajo:

- Emplear cámaras de mejor resolución, como por ejemplo de 16 megapíxeles, utilizar un elemento fijo donde realizar las capturas de las muestras, como por ejemplo un trípode. Tras esto crear una nueva la base de datos o añadir dichos datos a la actualmente existente, permitiendo sustituir o eliminar los datos con mayor ruido o simplemente añadir estos, ya que al aumenta el número de muestras la red puede realizar clasificaciones mejores.
- Aplicar un pre procesado, con el fin de disminuir ruido y homogeneizar las imágenes de la base de datos.
- Trabajar con las máquinas de soporte vectorial, con el fin de ver que técnica de clasificación da un mejor resultado, tanto en porcentaje de éxito como en coste computacional.
- Desarrollar una aplicación móvil, una vez que se disponga de un método de clasificación cuyo porcentaje de éxito sea superior al 95 %, para la identificación de rocas.
- Crear un código en bajo nivel que permita el procesado de los datos mucho más rápido, para una vez creada la aplicación móvil esta trabaje sobre dicho algoritmo, y permita reducir los tiempos de cómputo.
- Para tratar de reducir también el tiempo de cómputo se propone aplicar técnicas de reducción de la dimensionalidad de los parámetros de entrada.
- Realizar un artículo (*Paper's*) con los resultados obtenidos para la difusión de la información en aras de la investigación y desarrollo de I+D+i.

### 5.3. REFERENCIAS BIBLIOGRÁFICAS

- [1]. W. Griem. (2017). Apuntes Geología General, Métodos del reconocimiento de rocas [recurso web]. Recuperado de: <http://www.geovirtual2.cl/geologiageneral/ggcap03b.htm> [Consultado: 17/01/2018].
- [2]. Robert M. Haralick, K. Shanmugam, & Its'hak Dinstein. (1973). *Textural features for image classification* [versión electrónica]. *IEEE transactions on systems, man, and cybernetics*, vol. 3, Nº. 6. Recuperado de: <http://haralick.org/journals/TexturalFeatures.pdf>
- [3]. R. Gillespie. (1986). *LITHOLOGIC MAPPING OF SILICATE ROCKS USING TIMS* [versión electrónica]. California: *Institute of Technology, Pasadena*. Recuperado de: <https://ntrs.nasa.gov/search.jsp?R=19870007685>
- [4]. Sean Thompsona, Frank Fuetenb, David Bockusa. (2001). *Mineral identification using artificial neural networks and the rotating polarizer stage* [versión electrónica]. Canadá: *Department of Computer Science, University of Toronto & Department of Earth Sciences, Brock University*. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300400001539>
- [5]. Leena Lepistö, Iivari Kunttu, Jorma Autio and Ari Visa. (2002). *Comparison of some content-based image retrieval systems with rock texture images* [versión electrónica]. Finlandia: *Tampere University of Technology, Institute of Signal Processing*. Recuperado de: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.7786&rep=rep1&type=pdf>
- [6]. Roberto Marmoa, Sabrina Amodiob, Roberto Tagliaferri, Vittoria Ferrerib, Giuseppe Longof. (2005). *Textural identification of carbonate rocks by image processing and neural network: Methodology proposal and examples* [versión electrónica]. Italia: *Dipartimento di Informatica e Sistemistica, Università di Pavia, Dipartimento di Scienze della Terra, Università Federico II, Istituto per l'Ambiente Marino Costiero (IAMC), Geomare, National Research Council, Dipartimento Matematica e Informatica, Dipartimento di Scienze Fisiche, Università Federico II di Napoli, Università di Salerno & Istituto Nazionale di Fisica Nucleare*. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300404002493>
- [7]. Veerendra Singh, S. Mohan Rao (2005). *Application of image processing and radial basis neural network techniques for ore sorting and ore classification* [versión electrónica]. India: *Research and Development, Tata Steel, Jamshedpur*. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0892687505001081>
- [8]. J. Trinidad Guillen-Bonillaa, Evguenii Kurmyshev (2006). Estudio de la adyacencia en clases de imágenes de granito mediante la RICC [versión electrónica]. España: *Departamento de Metrología Óptica, Centro de Investigaciones en Óptica A. C., León*. Recuperado de: [http://congresos.cio.mx/3\\_enc\\_mujer/files/posters/Sesion%202/FMCT01.doc](http://congresos.cio.mx/3_enc_mujer/files/posters/Sesion%202/FMCT01.doc)
- [9]. Naresh Singh, T. N. Singh, Avyaktanand Tiwary. (2009). *Textural identification of basaltic rocks mass using processing images and neuronal network* [versión electrónica]. India: *Department of Mining Engineering, Institute of Technology, Banaras Hindu University & Department of Earth Sciences Indian Institute of Technology, Bombay*. Recuperado de: <https://link.springer.com/article/10.1007%2Fs10596-009-9154-x?LI=true>

- [10]. Nurdan Akhan Baykan, Nihat Yılmaz. (2009). *Mineral identification using color spaces and artificial neural networks* [version electrónica]. Turkia: Department of Computer Engineering & Department of Electric–Electronics Engineering, University of Selcuk. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300409002155>
- [11]. Claudio A. Pérez, Pablo A. Estévez, Pablo A. Vera, Luis E. Castillo, Carlos M. Aravena, Daniel A. Schulz, Leonel E. Medina (2011). *Ore grade estimation by feature selection and voting using boundary detection in digital image analysis* [version electrónica]. Chile: Department of Electrical Engineering and Advanced Mining Technology Center, Universidad de Chile. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0301751611001037>
- [12]. N. Yesiloglu-Gultekin, A.S.Keceli, E.A.Sezer, A.B.Can, C.Gokceoglu, H.Bayhan. (2012). *A computer program (TSecSoft) to determine mineral percentages using photographs obtained from thin sections* [versión electrónica]. Turkia: Department of Geological Engineering & Computer Engineering, Hacettepe University. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S0098300412000039>
- [13]. Snehamoy Chatterjee. (2012). *Vision-based rock-type classification of limestone using multiclass support vector machine* [Versión electrónica]. India: Dept. of Mining Engineering National Institute of Technology Rourkela, Orissa. Recuperado de: <https://link.springer.com/article/10.1007/s10489-012-0391-7>
- [14]. Mariusz Młynarczuk, Andrzej Górszczyk, Bartłomiej Ślipek. (2013). *The application of pattern recognition in the automatic classification of microscopic rock images* [version electrónica]. Polonia: University of Science and Technology, Faculty of Geology, Geophysics and Environmental Protection. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300413002112>
- [15]. Hossein Izadi, JavadSadri, Nosrat-Agha Mehran. (2015). *A new intelligent method for minerals segmentation in thin sections based on a novel incremental color clustering* [version electrónica]. Iran: Petroleum Exploration Department, Faculty of Mining Engineering, University College of Engineering, University of Tehran, Department of Computer Engineering, Faculty of Electrical and Computer Engineering & Department of Mining Engineering, Faculty of Engineering, University of Birjand. Recuperado de: <https://link.springer.com/article/10.1007/s10489-012-0391-7>
- [16]. Saeed Aligholi, RezaKhajavi, MortezaRazmara. (2015). *Automated mineral identification algorithm using optical properties of crystals* [version electrónica]. Iran: Department of Geology, Ferdowsi & Earthquake Research Center, University of Mashhad. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300415300558>
- [17]. Hossein Izadia, Javad Sadric, Mahdokht Bayatie. (2016). *An intelligent system for mineral identification in thin sections based on a cascade approach* [version electrónica]. Iran: Petroleum Exploration Department, Faculty of Mining Engineering, University College of Engineering, Institute of Geophysics, University of Tehran, Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Department of Mining Engineering & Faculty of Engineering University of Birjand. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0098300416305702>

- [18]. César Troya Sherdek. (2015). *LBP y ULBP – Local Binary Patterns y Uniform Local Binary Patterns* [recurso web]. Recuperado de: <https://cesartroyasherdek.wordpress.com/2016/02/26/deteccion-de-objetos-vi/> [Consultado: 17/01/2018].
- [19]. Ricardo J. Colom, Rafael Gadea, Ángel Sebastián, Marcos Martínez, Vicente Herrero, Vicente Arnau. (2001). *Transformada Discreta Wavelet 2-D para procesamiento de video en tiempo real* [versión electrónica]. Recuperado de: [https://www.uv.es/VARNAU/jor\\_pal\\_2001.pdf](https://www.uv.es/VARNAU/jor_pal_2001.pdf) [Consultado: 17/01/2018].
- [20]. Yepes Piqueras, Víctor. (2017) ¿Qué es y para qué sirve una red neuronal artificial? [recurso web]. Recuperado de: <http://victoryepes.blogs.upv.es/2017/01/07/que-es-y-para-que-sirve-una-red-neuronal-artificial/> [Consultado: 17/01/2018].
- [21]. Wikipedia. (2018). Red neuronal artificial [recurso web]. Recuperado de: [https://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](https://es.wikipedia.org/wiki/Red_neuronal_artificial) [Consultado: 17/01/2018].
- [22]. Ashok Kumar Patel, Snehamoy Chatterjee. (2016). *Computer vision-based limestone rock-type classification using probabilistic neural network* [versión electrónica]. USA: Department of Geological and Mining Engineering and Sciences, Michigan Technological University. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S1674987114001388>
- [23]. R. Blom, M. Daily. (1981). *Radar image processing for rock type discrimination* [versión electrónica]. DC Washington: IEEE IGARSS Tech. Dig. Recuperado de: <http://ieeexplore.ieee.org/abstract/document/4157309/>
- [24]. Li Wang. (1995). *Automatic identification of rocks in thin sections using texture analysis* [version electrónica]. Canadá: Centre d'applications et de recherches en télédétection (CARTEL) Université de Sherbrooke. Recuperado de: <https://link.springer.com/article/10.1007%2FBF02087099?LI=true>
- [25]. Marijke F. Augusteijn, Laura E. Clemens, and Kelly A. Shaw. (1995). *Performance Evaluation of Texture Measures for Ground Cover Identification in Satellite Images by Means of a Neural Network Classifier*. Recuperado de: <http://ieeexplore.ieee.org/abstract/document/387577/>
- [26]. Ivan Vasilev (2014). Un Tutorial de Aprendizaje Profundo: De Perceptrones a Redes Profundas [recurso web]. Recuperado de: <https://www.toptal.com/machine-learning/un-tutorial-de-aprendizaje-profundo-de-perceptrones-a-redes-profundas/es>. [Consultado: 12/01/2018].
- [27]. Guillermo Julián (2014). Las redes neuronales: qué son y por qué están volviendo [recurso web]. Recuperado de: <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>. [Consultado: 12/01/2018].
- [28]. Pantechsolutions. (2015). *Matlab Code for Discrete Wavelet Transform* [recurso web]. Recuperado de: <https://www.pantechsolutions.net/blog/matlab-code-for-discrete-wavelet-transform/>
- [29]. Adrian Rosebrock. (2015). *Local Binary Patterns with Python & OpenCV* [recurso web]. Recuperado de: <https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>. [Consultado: 12/01/2018].
- [30]. MathWorks. Familias Wavelet [recurso web]. Recuperado de: <https://es.mathworks.com/help/wavelet/ref/waveletfamilies.html>. [Consultado: 11/01/2018].
- [31]. MathWorks. DWT [recurso web]. Recuperado de: <https://es.mathworks.com/help/wavelet/ref/dwt.html> [Consultado: 11/01/2018].



# ANEXO A: RESUMEN DEL ESTADO DEL ARTE



Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas?	Programa empleado
<b>Textural features for image classification</b>	1973	IEEE TRANSACTION S ON SYSTEMS, MAN, AND CYBERNETICS	ROBERT M. HARALICK, K. SHANMUGAM , AND ITS'HAK DINSTEIN	Textural (Vecino más próximo - función de discriminación lineal)	Arenisca	243	89	Rocas arenisca, Imágenes satelitales y fotogramétricas	N/S
<b>Automatic Identification of Rocks in Thin Sections Using Texture Analysis</b>	1995	Mathematical Geology, Vol. 27. No. 7	Li Wang	Algoritmo de clasificación supervisada y reglas de decisión de mínima distancia (Matriz de co-ocurrencia, textura espectral y espaciado textural)	Metamórficas [ <i>mylonite (M)</i> ] - Ígneas [ <i>diorite porphyry (DP), diorite (D), gabbro, (Ga), granite (Gr), and periodotite (P)</i> ]. ( <i>Ga</i> ), <i>granite (Gr)</i> , and <i>periodotite (P)</i> .	58	89	Secciones finas de rocas de montaña Debie al este de China	N/S
<b>Mineral identification using artificial neural networks and</b>	2001	Computers & Geosciences	Sean Thompsona, Frank Fuetenb, David Bockusa	ANN - Microscopio de luz polarizada	Ígnea, Metamórfica y sedimentaria (16)	44 - [Entrenamiento de la red, 36 y 74 muestras por mineral]	93	Secciones finas de rocas con luz polarizada	N/S

Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas?	Programa empleado
<i>the rotating polarizer stage.</i>									
<i>Textural identificación of carbonate rocks bu image processing and neural network: Methodology proposal and examples.</i>	2005	<i>Computers &amp; Geosciences</i>	<i>Roberto Marmoa, , Sabrina Amodiob, Roberto Tagliaferrid, Vittoria Ferrerib, Giuseppe Longof.</i>	<i>ANN</i>	<i>Carbonatadas</i>	<i>&gt; 1000, 2 entrenamientos de la red, una con 256 y otra con 215 de imágenes de distintas zonas</i>	<i>93,3-93,5</i>	<i>Secciones finas de rocas marinas</i>	<i>Matlab</i>
<i>Application of image processing and radial basis neural network techniques for ore sorting and ore classification</i>	2005	<i>ELSERVIER - Minerals Engineering</i>	<i>Veerendra Singh, S. Mohan Rao</i>	<i>ANN</i>	<i>* (Planta metalúrgica ferromanganeso)</i>	<i>3 imágenes de minerales enriquecidos( Manganeso, Hierro, Alúmina, Agregados), de los tres minerales para identificar, empleando</i>	<i>88,71</i>	<i>Clasifica minerales de las imágenes RGB (Manganeso, Hierro, Alúmina, Agregados)</i>	<i>Matlab</i>

Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas?	Programa empleado
						las estadísticas de las misma en el entrenamiento de la red.			
<b>Estudio de la adyacencia en clase de imágenes de granito mediante RICC. DE GRANITO MEDIANTE LA RICC</b>	2006	III Encuentro de la mujer en la ciencia	J. Trinidad Guillen-Bonillaa, Evguenii Kurmyshev.	Representación de Imágenes por Cúmulos Coordinados (RICC) - Clasificador por distancia mínima	Granito	36 imágenes, 8 clases, 3 imágenes por clase	N/S	Muestras de granitos ornamentales, Rosa Porriño	N/S
<i>Mineral identification using color spaces and artificial neural networks.</i>	2009	<i>Computers &amp; Geosciences</i>	<i>Nurdan Akhan Baykan, Nihat Yilmaz</i>	<i>ANN - Microscopio de luz polarizada - Cruzada</i>	Minerales	390 imágenes de distinta iluminación y ángulo de captura	81-98	Identificación de minerales (namely, quartz, muscovite, biotite, chlorite, and opaque)	Matlab
<i>Textural identification of basaltic rocks mass</i>	2009	<i>Computers &amp; Geosciences</i>	<i>Naresh Singh · T. N. Singh · Avyaktanand Tiwary ·</i>	<i>ANN</i>	Basáltica	300 secciones finas, 140 muestras de rocas	92,22	Clasificación de texturas de rocas basálticas	Matlab

Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas?	Programa empleado
<i>using processing images and neuronal network.</i>			<i>Naresh Singh, T. N. Singh, Avyaktanand Tiwary, Kripa M. Sarkar</i>			basálticas, se emplearon para el entrenamiento de texturas, 90 secciones finas, 30 por cada tipo de textura a identificar (Intergranular , Porphyritic, Ophitic)			
<i>Ore grade estimation by feature selection and voting using boundary detection in digital image analysis.</i>	2011	ELSERVIER - Minerals Processing	<i>Claudio A. Perez, Pablo A. Estévez, Pablo A. Vera, Luis E. Castillo, Carlos M. Aravena, Daniel A. Schulz, Leonel E. Medina Daniel A. Schulz, Leonel E. Medina</i>	<i>Support Vector Machine (SVM)</i>	Níquel	Base de datos con 530 imágenes de rocas secas y húmedas (265), de la mina de Quebec - Canadá	RMSE (29,8%)	Clasificación del grado de níquel en las imágenes mediante clasificación de minerales (massive sulfide (MS), disseminated sulfide (DS), net textured (NT), gabbro (G), and peridotite	N/S

Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas? (P)).	Programa empleado
<b>A computer program (TSecSoft) to determine mineral percentages using photographs obtained from thin sections</b>	2012	<i>Computers &amp; Geosciences</i>	<i>N. Yesiloglu-Gultekin, A.S.Keceli, E.A.Sezer, A.B.Can, C.Gokceoglu, H.Bayhan</i>	<i>TSecSoft (Pre-procesado - Segmentación - Etiquetado)</i>	Granítica	Base de datos de la aplicación	N/S	Clasifica porcentajes de minerales de las secciones finas	Matlab
<b>Vision-based rock-type classification of limestone using multiclass support vector machine.</b>	2012	<i>Applied Intelligence</i>	<i>Snehamoy Chatterjee</i>	<i>Support Vector Machine (SVM)</i>	Calizas	1200 imágenes, 10 imágenes por muestra, con distintas orientaciones y lugar.	96,2	Identificación de rocas en la mina en la India Occidental	Matlab
<b>A new intelligent method for minerals segmentation</b>	2015	<i>Computers &amp; Geosciences</i>	<i>Hossein Izadi, JavadSadri, Nosrat-AghaMehran</i>	Algoritmo de agrupación por color (Segmentación mineral)	Ígneas	155 muestras de 28 minerales diferentes.	92,15-85,24	Identificación de rocas ígneas por agrupación de características	Matlab

Título	Año	Donde se público	Autores	Método de clasificación	Tipo de rocas	Número de muestras por tipo de roca	Precisiones (%)	¿Clasifica imágenes de rocas?	Programa empleado
<i>in thin sections based on a novel incremental color clustering.</i>				inteligente)				de color en las imágenes, y con muestras de distintos tipos de minerales.	
<b>Automated mineral identification algorithm using optical properties of crystals.</b>	2015	<i>Computers &amp; Geosciences</i>	<i>Saeed Aligholi, RezaKhajavi, MortezaRazmara</i>	<i>Majority Voting Schema (MVS)</i>	Ígneas	45 muestras de sección fina, 30 de plutónicas y 15 extrusivas e hiperbisal.	98	Identificación de minerales de rocas ígneas.	N/S
<b>An intelligent system for mineral identification in thin sections based on a cascade approach.</b>	2016	<i>Computers &amp; Geosciences</i>	<i>Hossein Izadia, Javad Sadric, Mahdokht Bayatie</i>	ANN-segmentación e identificación	Ígneas	135 imágenes de sección fina, 23 muestras de minerales de rocas ígneas,	93,81	Identificación de minerales de rocas ígneas	Matlab

# ANEXO B: ALGORITMOS EMPLEADOS



En se siguiente anexo se encuentran los códigos empleados para realizar el TFM, donde se crearon un total de 4 códigos propio (500 líneas) y se empleó un código suministrado por el tutor (230 líneas).

### Algoritmo para pasar imágenes a escala de grises.

```
% Algoritmo para pasar las imágenes de la base de datos a escala de grises y
% almacenados en otra carpeta, donde se clasificaran en imágenes para entrenar y para
%testear.
%----- Prueba Inicial -----%
% Se convierte la imagen a escala de grises.
% Se lee la imagen y la se visualiza.
%image = imread('C2_10.jpg');
%figure; imshow(rgb2gray(image))
% Se lee la imagen y se visualiza en binario.
%bw = im2bw(image);
%figure; imshow (bw)
%Se almacenan la imagen en un directorio distinto
%imwrite (BW, fullfile ('E: \TFM\BBDD_binario','C2_10_bw.jpg'));
%----- Algoritmo -----%
% Se cambia de directorio donde está la base de datos de las imágenes.
cd E:\TFM\black&White
% Se realiza la un doble bucle para la lectura y escritura de las imágenes.
% Clasificación de las imágenes de entrenamiento
for i=1:19
    for j=1:50
        filename = ['C', int2str(i), '_', int2str(j), '.jpg'];
        filename2 = ['Ct', int2str(i), '_', int2str(j), '.jpg'];
        image = imread (filename);
        if j <= 35
            gris = rgb2gray(image);
            imwrite(gris,fullfile('E:\TFM\black&White',filename));
        else
            gris = rgb2gray(image);
```

```

        imwrite(gris,fullfile('E:\TFM\black&White',filename2));
    end
end
end
% Clasificación de las imágenes de testeo
for i=1:19
    k =1 ;
    for j=36:50
        filename = ['Ct', int2str(i), '_', int2str(j), '.jpg'];
        images = imread(filename);
        filename3 =['Ct', int2str(i), '_', int2str(k), '.jpg'];
        imwrite(images,fullfile('E:\TFM\black&White',filename3));
        k = k+1;
    end
end
end

```

### Algoritmo para obtener LBP

Este algoritmo ha sido suministrado por el tutor del TFM.

*%LBP returns the local binary pattern image or LBP histogram of an image.*

*% J = LBP(I,R,N,MAPPING,MODE) returns either a local binary pattern*

*% coded image or the local binary pattern histogram of an intensity*

*% image I. The LBP codes are computed using N sampling points on a*

*% circle of radius R and using mapping table defined by MAPPING.*

*% See the getmapping function for different mappings and use 0 for*

*% no mapping. Possible values for MODE are*

*% 'h' or 'hist' to get a histogram of LBP codes*

*% 'nh' to get a normalized histogram*

*% Otherwise an LBP code image is returned.*

*%*

*% J = LBP(I) returns the original (basic) LBP histogram of image I*

*%*

*% J = LBP(I,SP,MAPPING,MODE) computes the LBP codes using n sampling*

```

% points defined in (n * 2) matrix SP. The sampling points should be
% defined around the origin (coordinates (0,0)).
%
% Examples
% -----
% I=imread('rice.png');
% mapping=getmapping(8,'u2');
% H1=LBP(I,1,8,mapping,'h'); %LBP histogram in (8,1) neighborhood
%
%           %using uniform patterns
% subplot(2,1,1),stem(H1);
%
% H2=LBP(I);
% subplot(2,1,2),stem(H2);
%
% SP=[-1 -1; -1 0; -1 1; 0 -1; 0 1; 1 -1; 1 0; 1 1];
% I2=LBP(I,SP,0,'i'); %LBP code image using sampling points in SP
%
%           %and no mapping. Now H2 is equal to histogram
%           %of I2.

function result = lbp(varargin) % image,radius,neighbors,mapping,mode)
% Version 0.3.3
% Authors: Marko Heikkilä and Timo Ahonen
% Changelog
% Version 0.3.2: A bug fix to enable using mappings together with a
% predefined spoints array
% Version 0.3.1: Changed MAPPING input to be a struct containing the mapping
% table and the number of bins to make the function run faster with high number
% of sampling points. Lauge Sorensen is acknowledged for spotting this problem.
% Check number of input arguments.
error(nargchk(1,5,nargin));
image=varargin{1};
d_image=double(image);

```

```
if nargin==1
    spoints=[-1 -1; -1 0; -1 1; 0 -1; -0 1; 1 -1; 1 0; 1 1];
    neighbors=8;
    mapping=0;
    mode='h';
end
if (nargin == 2) && (length(varargin{2}) == 1)
    error('Input arguments');
end
if (nargin > 2) && (length(varargin{2}) == 1)
    radius=varargin{2};
    neighbors=varargin{3};
    spoints=zeros(neighbors,2);
    % Angle step.
    a = 2*pi/neighbors;
    for i = 1:neighbors
        spoints(i,1) = -radius*sin((i-1)*a);
        spoints(i,2) = radius*cos((i-1)*a);
    end
if(nargin >= 4)
    mapping=varargin{4};
    if(isstruct(mapping) && mapping.samples ~= neighbors)
        error('Incompatible mapping');
    end
else
    mapping=0;
end
if(nargin >= 5)
    mode=varargin{5};
else
    mode='h';
end
```

```

end
if (nargin > 1) && (length(varargin{2}) > 1)
    spoints=varargin{2};
    neighbors=size(spoints,1);
    if(nargin >= 3)
        mapping=varargin{3};
        if(isstruct(mapping) && mapping.samples ~= neighbors)
            error('Incompatible mapping');
        end
    else
        mapping=0;
    end
    if(nargin >= 4)
        mode=varargin{4};
    else
        mode='h';
    end
end
end
% Determine the dimensions of the input image.
[ysize xsize] = size(image);
miny=min(spoints(:,1));
maxy=max(spoints(:,1));
minx=min(spoints(:,2));
maxx=max(spoints(:,2));
% Block size, each LBP code is computed within a block of size bsizey*bsizex
bsizey=ceil(max(maxy,0))-floor(min(miny,0))+1;
bsizex=ceil(max(maxx,0))-floor(min(minx,0))+1;
% Coordinates of origin (0,0) in the block
origy=1-floor(min(miny,0));
origx=1-floor(min(minx,0));
% Minimum allowed size for the input image depends
% on the radius of the used LBP operator.

```

```

if(xsize < bsize || ysize < bsizey)
    error('Too small input image. Should be at least (2*radius+1) x (2*radius+1)');
end
% Calculate dx and dy;
dx = xsize - bsize;
dy = ysize - bsizey;
% Fill the center pixel matrix C.
C = image(origy:origy+dy,origx:origx+dx);
d_C = double(C);
bins = 2^neighbors;
% Initialize the result matrix with zeros.
result=zeros(dy+1,dx+1);

%Compute the LBP code image

for i = 1:neighbors
    y = spoints(i,1)+origy;
    x = spoints(i,2)+origx;
    % Calculate floors, ceils and rounds for the x and y.
    fy = floor(y); cy = ceil(y); ry = round(y);
    fx = floor(x); cx = ceil(x); rx = round(x);
    % Check if interpolation is needed.
    if (abs(x - rx) < 1e-6) && (abs(y - ry) < 1e-6)
        % Interpolation is not needed, use original datatypes
        N = image(ry:ry+dy,rx:rx+dx);
        D = N >= C;
    else
        % Interpolation needed, use double type images
        ty = y - fy;
        tx = x - fx;

        % Calculate the interpolation weights.

```

```

w1 = roundn((1 - tx) * (1 - ty),-6);
w2 = roundn(tx * (1 - ty),-6);
w3 = roundn((1 - tx) * ty,-6) ;
% w4 = roundn(tx * ty,-6) ;
w4 = roundn(1 - w1 - w2 - w3, -6);
% Compute interpolated pixel values
N = w1*d_image(fy:fy+dy,fx:fx+dx) + w2*d_image(fy:fy+dy,cx:cx+dx) + ...
w3*d_image(cy:cy+dy,fx:fx+dx) + w4*d_image(cy:cy+dy,cx:cx+dx);
N = roundn(N,-4);
D = N >= d_C;
end
% Update the result matrix.
v = 2^(i-1);
result = result + v*D;
end
%Apply mapping if it is defined
if isstruct(mapping)
    bins = mapping.num;
    for i = 1:size(result,1)
        for j = 1:size(result,2)
            result(i,j) = mapping.table(result(i,j)+1);
        end
    end
end
end
if (strcmp(mode,'h') || strcmp(mode,'hist') || strcmp(mode,'nh'))
    % Return with LBP histogram if mode equals 'hist'.
    result=hist(result(:,0):(bins-1));
    if (strcmp(mode,'nh'))
        result=result/sum(result);
    end
end
else
    %Otherwise return a matrix of unsigned integers

```

```
if ((bins-1)<=intmax('uint8'))
    result=uint8(result);
elseif ((bins-1)<=intmax('uint16'))
    result=uint16(result);
else
    result=uint32(result);
end
end
end
function x = roundn(x, n)
error(nargchk(2, 2, nargin, 'struct'))
validateattributes(x, {'single', 'double'}, {}, 'ROUNDN', 'X')
validateattributes(n, ...
    {'numeric'}, {'scalar', 'real', 'integer'}, 'ROUNDN', 'N')
if n < 0
    p = 10 ^ -n;
    x = round(p * x) / p;
elseif n > 0
    p = 10 ^ n;
    x = p * round(x / p);
else
    x = round(x);
end
end
```

### Algoritmo creado para el primer experimento

% Algoritmo TFM reconocimiento de rocas a partir de imágenes de

% dispositivos móviles.

% Nº Total de clases -> 19

% Nº Total de muestras por clases -> 50

% Nº de muestras para entrenamiento 70% -> 35

```

% Nº de muestras para test 30% -> 15
% Se define donde se almacenaran los parámetros por muestras de
% entrenamiento así como la etiqueta que le corresponde por clase.
clear;clc;
% -----Entrenamiento----- %
param=zeros(256,665);
target=zeros(19,665);
k=1;
for i=1:19
    clase = i;
    for j=1:35
        filename = imread(['C', int2str(i), '_', int2str(j), '.jpg']);
        LBP =lbp(filename);

        %Se almacenan los parámetros dados por el algoritmo del local binary
        %parttern
        for m = 1:256
            LBP_1 = LBP(1, m);
            param(k,m) =LBP_1;
        end
        k=k+1;
    end
end
end
%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19
    for j=1:35
        target(i, j+E) =1;
    end
    E =E+j;
end
end
% Se define donde se almacenaran los parámetros por muestras de

```

```
% testeo así como la etiqueta que le corresponde por clase.
% -----Testeo-----%
param2=zeros(256,285);
target2=zeros(19,285);
k=1;
for i=1:19
    clase = i;
    for j=1:15
        filename = imread(['Ct', int2str(i), '_', int2str(j), '.jpg']);
        LBP =lbp(filename);
%Se almacenan los parámetros dados por el algoritmo del local binary
%parttern
        for m = 1:256
            LBP_1 = LBP(1, m);
            param2(k,m) =LBP_1;
        end
        k=k+1;
    end
end
%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19
    for j=1:15
        target2(i, j+E) =1;
    end
    E =E+j;
end
% Se realiza la red neuronal con los datos obtenidos.
% Normalización de datos
parametrosTraining=param;
parametrosTest=param2;
[pn,ps] = mapminmax(parametrosTraining);
```

```
[tn,ts] = mapminmax('apply',parametrosTest,ps);
neuronasocultas=24;
%creación de arquitectura
net = newff(pn,target,neuronasocultas);
%Entrenamiento
net = train(net,pn,target);
%Testeo
etiquetasalida=sim(net,tn);
% Se definen nuestra matriz de confusión
[porcentaje_fallo, matriz_confusion] = confusion(target2, etiquetasalida);
porcentaje_acierto = 1-porcentaje_fallo;
%Sobre el algoritmo anterior se aplica una variante en el entrenamiento de la red, donde
%se calcula la red un número determinado de veces, esto es necesario ya que la red toma
%datos aleatorios en la primera asignación de pesos, siendo necesario realizar sobre la
%red varias iteraciones, con el fin de disminuir el error y aproximar al valor real, para ello
%se modifica el algoritmo en la creación de la red.
% Mediante bucle for se entrena la red
etiquetasalida_0=zeros(19,285);
neuronasocultas=24;
for x=1:10
    %creación de arquitectura
    net = newff(pn,target,neuronasocultas);
    %Entrenamiento
    net = train(net,pn,target);
    %Testeo
    etiquetasalida=sim(net,tn);
    etiquetasalida_0=etiquetasalida_0 + etiquetasalida;
end
etiquetasalida=etiquetasalida_0/x;
% Se definen nuestra matriz de confusión
[porcentaje_fallo, matriz_confusion] = confusion(target2, etiquetasalida);
porcentaje_acierto = 1-porcentaje_fallo;
```

## Algoritmo creado para el segundo experimento

```

%Algoritmo para obtener dwt – LBP por nivel DWT
% N° Total de clases -> 19
% N° Total de muestras por clases -> 50
% N° de muestras para entrenamiento 70% -> 35
% N° de muestras para test 30% -> 15
% N° de veces que se hace el DWT en la imagen -> 4
% Se definen nuestra donde se almacenaran los parámetros por muestras de
% entrenamiento así como la etiqueta que le corresponde por clase.
clear;clc;
% -----Entrenamiento----- %
param=zeros(2660,256);
target=zeros(19,2660);
k=1;
for i=1:19
    for j=1:35
        filename = imread(['C', int2str(i), '_', int2str(j), '.jpg']);
        % Se usa el DWT sobre la imagen
        [cA,cH,cV,cD] = dwt2(filename, 'sym4');
        % Se usa el DWT sobre el filtro de paso bajo, hasta 3 veces.
        [cA2,cH2,cV2,cD2] = dwt2(cA, 'sym4');
        [cA3,cH3,cV3,cD3] = dwt2(cA2, 'sym4');
        [cA4,cH4,cV4,cD4] = dwt2(cA3, 'sym4');
        % Se aplica el LBP sobre las cA y se concatenan.
        LBP =[lbp(cA); lbp(cA2); lbp(cA3); lbp(cA4)];
        %Normalización de características.
        %Se almacenan los parámetros dados por el algoritmo del local binary
        %parttern
        for r =1:4
            for m = 1:256
                LBP_1 = LBP(r, m);
            end
        end
    end
end

```

```

        param(k,m) =LBP_1;
    end
    k=k+1;
end
end
end
%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19
    for j=1:140
        target(i, j+E) =1;
    end
    E =E+j;
end
% -----Testeo----- %
param2=zeros(1140,256);
target2=zeros(19,1140);
k=1;
for i=1:19
    for j=1:15
        filename = imread(['Ct', int2str(i), '_', int2str(j), '.jpg']);
        % Se usa el DWT sobre la imagen
        [cA,cH,cV,cD] = dwt2(filename, 'sym4');
        % Se usa el DWT sobre el paso bajo bajo, hasta 3 veces.
        [cA2,cH2,cV2,cD2] = dwt2(cA, 'sym4');
        [cA3,cH3,cV3,cD3] = dwt2(cA2, 'sym4');
        [cA4,cH4,cV4,cD4] = dwt2(cA3, 'sym4');
        % Se aplica el LBP sobre las cA y se concatenan.
        LBP =[lbp(cA); lbp(cA2); lbp(cA3); lbp(cA4)];
        %Se almacenan los parámetros dados por el algoritmo del local binary
        %parttern
    for r =1:4

```

```

    for m = 1:256
        LBP_1 = LBP(r, m);
        param2(k,m) =LBP_1;
    end
    k=k+1;
end
end
end
end
%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19
    for j=1:60
        target2(i, j+E) =1;
    end
    E =E+j;
end
end

```

### Algoritmo creado para el tercer experimento

```

% Se realiza la red neuronal con los datos obtenidos.
% Normalización de datos
%Se debe tener en cuenta que el número de filas ha de ser el mismo en la red, por
%lo tanto se aplica la transpuesta a los parámetros obtenidos.
parametrosTraining=param';
parametrosTest=param2';
[pn,ps] = mapminmax(parametrosTraining);
[tn,ts] = mapminmax('apply',parametrosTest,ps);
% neuronasocultas = 24;
% %creación de arquitectura
% net = newff(pn,target,neuronasocultas);
% %Entrenamiento
% net = train(net,pn,target);

```

```

%%Testeo
% etiquetasalida=sim(net,tn);
% [porcentaje_fallo, matriz_confusion] = confusion(target2, etiquetasalida);
% porcentaje_acierto = 1-porcentaje_fallo;
% Mediante bucle for se entrena la red
etiquetasalida_0=zeros(19,1140);
neuronasocultas=10;
for x=1:12
    %creación de arquitectura
    net = newff(pn,target,neuronasocultas);
    %Entrenamiento
    net = train(net,pn,target);
    %Testeo
    etiquetasalida=sim(net,tn);
    etiquetasalida_0=etiquetasalida_0 + etiquetasalida;
end
etiquetasalida=etiquetasalida_0/x;
% Se definen nuestra matriz de confusión
[porcentaje_fallo, matriz_confusion] = confusion(target2, etiquetasalida);
porcentaje_acierto = 1-porcentaje_fallo;
El algoritmo creado es el siguiente:
% ALGORITMO LBP y DWT CONJUNTAMENTE
% Nº Total de clases -> 19
% Nº Total de muestras por clases -> 50
% Nº de muestras para entrenamiento 70% -> 35
% Nº de muestras para test 30% -> 15
% Nº de veces que se hace el DWT en la imagen -> 4
% Se definen nuestra donde se almacenaran los parámetros por muestras de
% entrenamiento así como la etiqueta que le corresponde por clase.
clear;clc;
% -----Entrenamiento----- %
param=zeros(256,665);

```

```

target=zeros(19,665);
k=1;
for i=1:19
    clase = i;
    for j=1:35
        filename = imread(['C', int2str(i), '_', int2str(j), '.jpg']);
        filename = imread('C1_1.jpg');
        % Se usa el DWT sobre la imagen
        [cA,cH,cV,cD] = dwt2(filename, 'haar');
        %' Una vez Se calcula se prueba con las otras dos familias bior4.4', 'sym4'.
        % Se usa el DWT sobre el filtro de paso bajo bajo, hasta 3 veces.
        [cA2,cH2,cV2,cD2] = dwt2(cA, 'haar');
        [cA3,cH3,cV3,cD3] = dwt2(cA2, 'haar');
        [cA4,cH4,cV4,cD4] = dwt2(cA3, 'haar');
        % Se aplica el LBP sobre las cA y se concatenan, con el fin de homogeneizar los datos al
        % tratarse de imágenes de distinto tamaño.
        LBP =[lbp(cA), lbp(cA2), lbp(cA3), lbp(cA4)];
        %Se almacenan los parámetros dados por el algoritmo del local binary
        %parttern
        for m = 1:1024
            LBP_1 = LBP(1, m);
            param(k,m) =LBP_1;
        end
        k=k+1;
    end
end
%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19
    for j=1:35
        target(i, j+E) =1;
    end
end

```

```

E =E+j;
end
% -----Testeo-----%
param2=zeros(256,285);
target2=zeros(19,285);
k=1;
for i=1:19
    clase = i;
    for j=1:15
        filename = imread(['Ct', int2str(i), '_', int2str(j), '.jpg']);
        % Se usa el DWT sobre la imagen
        [cA,cH,cV,cD] = dwt2(filename, 'haar');
        %' Una vez se calcula se prueba con las otras dos familias bior4.4', 'sym4'.
        % Se usa el DWT sobre el paso bajo bajo, hasta 3 veces.
        [cA2,cH2,cV2,cD2] = dwt2(cA, ' haar');
        [cA3,cH3,cV3,cD3] = dwt2(cA2, ' haar');
        [cA4,cH4,cV4,cD4] = dwt2(cA3, ' haar');
        % Se aplica el LBP sobre las cA y se concatenan.
        LBP =[lbp(cA), lbp(cA2), lbp(cA3), lbp(cA4)];
        %Se almacenan los parámetros dados por el algoritmo del local binary
        %parttern
        for m = 1:1024
            LBP_1 = LBP(1, m);
            param2(k,m) =LBP_1;
        end
        k=k+1;
    end
end
end

%Se etiqueta la matriz de etiquetas target
E = 0;
for i =1:19

```

```
for j=1:15
    target2(i, j+E) =1;
end
E =E+j;
end
% Se realiza la nuestra red neuronal con los datos obtenidos.
% Normalización de datos
parametrosTraining=param1;
parametrosTest=param2';
[pn,ps] = mapminmax(parametrosTraining);
[tn,ts] = mapminmax('apply',parametrosTest,ps);

% Mediante bucle for se entrena la red
etiquetasalida_0=zeros(19,285);
neuronasocultas=10;
for x=1:10
    %creación de arquitectura
    net = newff(pn,target,neuronasocultas);
    %Entrenamiento
    net = train(net,pn,target);
    %Testeo
    etiquetasalida=sim(net,tn);
    etiquetasalida_0=etiquetasalida_0 + etiquetasalida;
end
etiquetasalida=etiquetasalida_0/x;
% Se definen nuestra matriz de confusión
[porcentaje_fallo, matriz_confusion] = confusion(target2, etiquetasalida);
porcentaje_acierto = 1-porcentaje_fallo;
```



